CMPG-767 Image Processing and Analysis

FREQUENCY DOMAIN FILTERING. PERIODIC NOISE FILTERING

Frequency Domain Filtering

- Filtering in the frequency domain consists of modifying the Fourier transform of an image and then calculating the inverse transform to obtain the processed result

$$\hat{f}(x,y) = F^{-1}\Big(F\big(g(x,y)\big)H(u,v)\Big) = F^{-1}\Bigg(\underbrace{G(u,v)}_{F(g(x,y))}H(u,v)\Bigg)$$

where $H(u,v)$ is a filter-function in the frequency domain

$G(u,v) = F\big(g(x,y)\big)$ is a Fourier transform of the image to be filtered and H can be treated as a Fourier transform of some kernel h

Frequency Domain Filtering

- This concept of frequency domain filtering

$$\hat{f}(x,y) = F^{-1}\left(F\left(g(x,y)\right)H(u,v)\right) = F^{-1}\left(\underbrace{G(u,v)}_{F(g(x,y))}H(u,v)\right)$$

- matches the concept of linear spatial domain filtering where filtering is reduced to the convolution with a filtering kernel $h(x, y)$ $\hat{f}(x,y) = g(x,y) * h(x,y)$

- In fact, according to the Convolution Theorem

$$\hat{f}(x,y) = F^{-1}\left(G(u,v)\underbrace{H(u,v)}_{F(h(x,y))}\right) = g(x,y) * h(x,y)$$

Frequency Domain Filtering

- **Any** linear filter in the spatial domain (which we considered earlier) can be utilized in the frequency domain because any local neighborhood convolution with the same kernel window W can be utilized through a global convolution of a noisy image with a corresponding convolutional kernel $\hat{f}(x,y) = g(x,y) * h(x,y)$
- where

$$W = \begin{pmatrix} w_{11} & \dots & \dots & w_{1m} \\ w_{21} & \dots & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{n1} & \dots & \dots & w_{nm} \end{pmatrix} \qquad h = \begin{pmatrix} [W] & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} w_{11} & \dots & w_{1m} & 0 & \dots & 0 & 0 \\ w_{21} & \dots & w_{2m} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & 0 & \dots & 0 & 0 \\ w_{n1} & \dots & w_{nm} & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Frequency Domain Filtering

- For example, let us consider smart linear filter with a 3x3 kernel

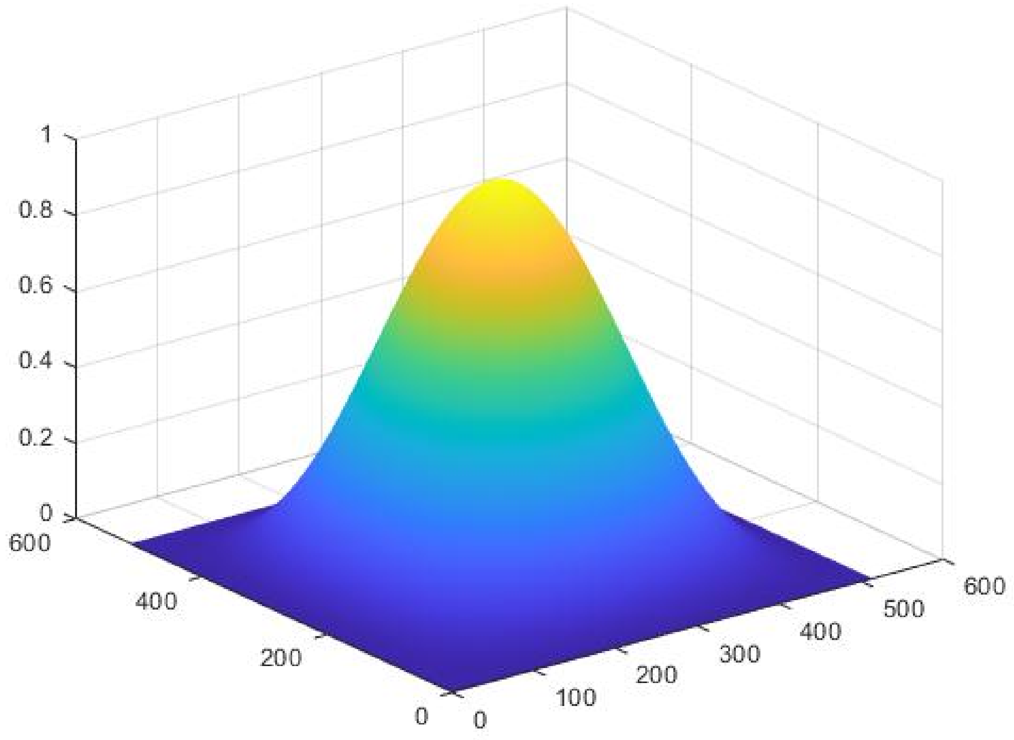
$$W = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{21} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 0.0625 & 0.1250 & 0.0625 \\ 0.1250 & 0.2500 & 0.1250 \\ 0.0625 & 0.1250 & 0.0625 \end{pmatrix}$$

- Then

$$h = \begin{pmatrix} [W] & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0.0625 & 0.125 & 0.0625 & 0 & \dots & 0 \\ 0.125 & 0.25 & 0.125 & 0 & \dots & 0 \\ 0.0625 & 0.125 & 0.0625 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Frequency Domain Filtering

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to smart linear filter:

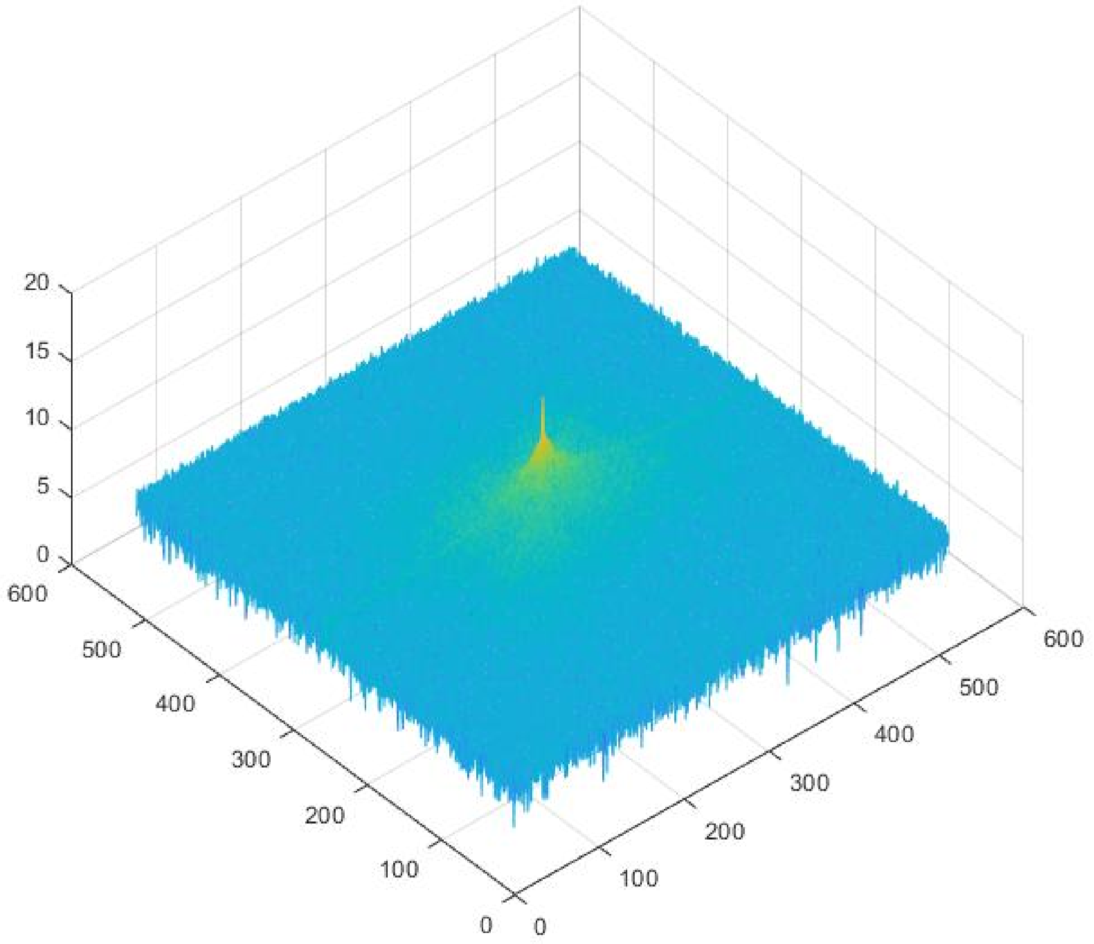


Frequency Domain Filtering

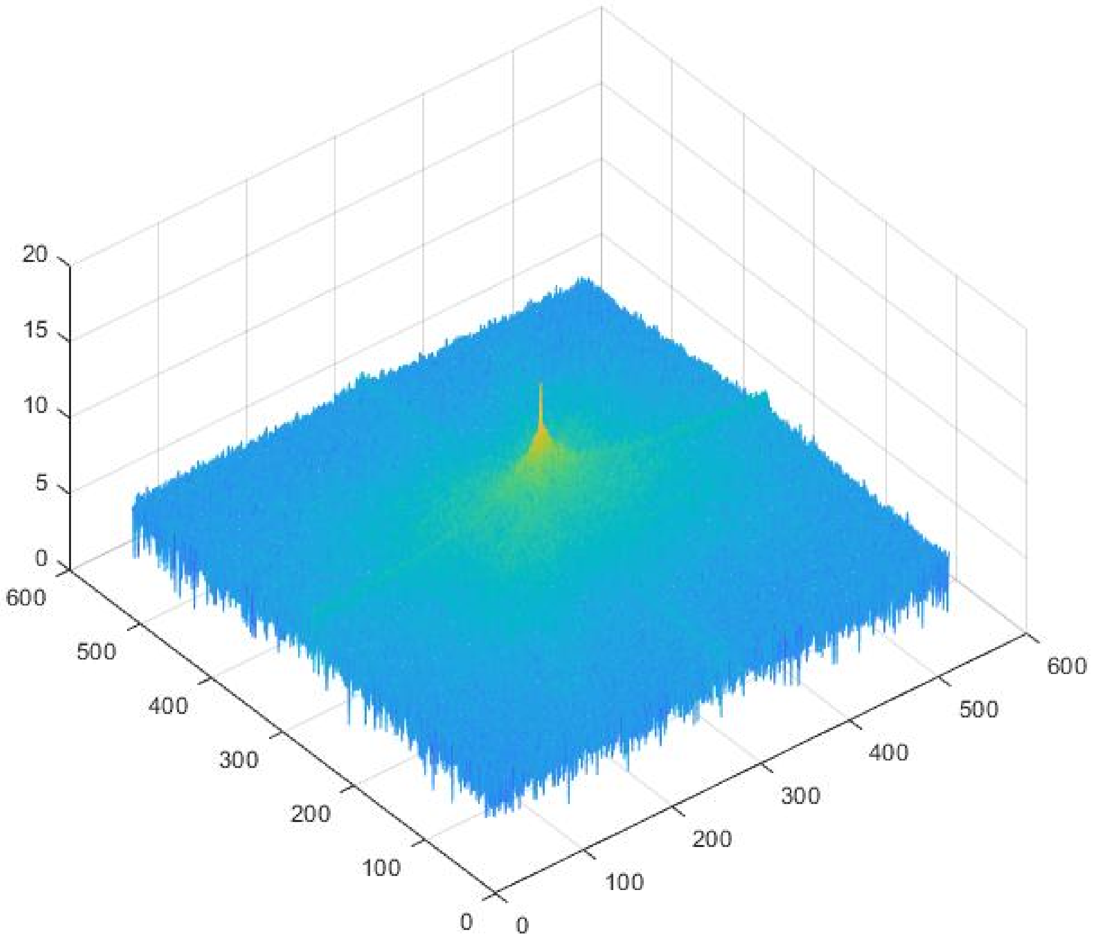
Original “Lena” (image f)

“Lena” corrupted with additive Gaussian noise 0.2σ (image g)

Frequency Domain Filtering



Fourier Power spectrum $|F|$ of “Lena”

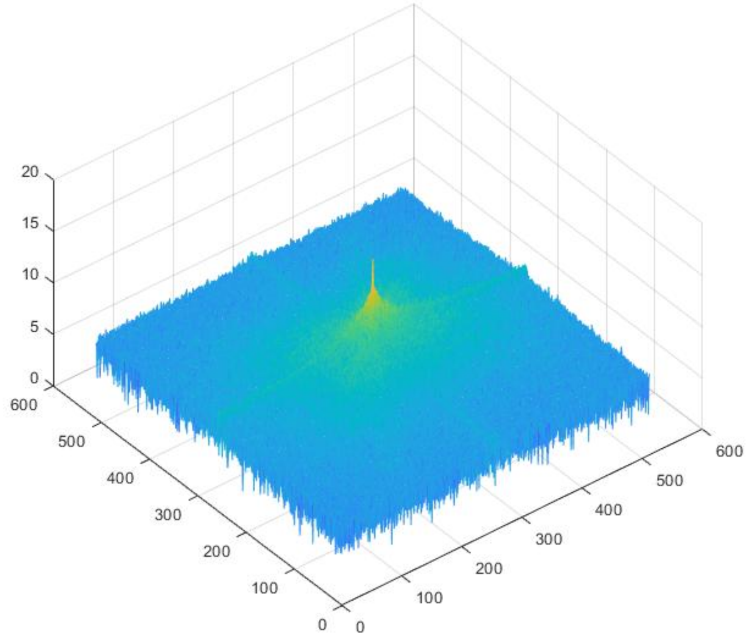


Fourier Power spectrum $|G|$ of noisy “Lena”

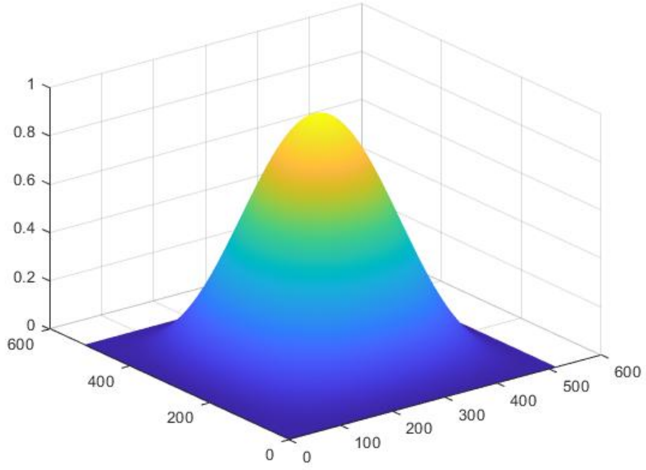
Frequency Domain Filtering

- According to the convolution theorem a component-wise product of Fourier transforms G of an image g and H of a convolutional kernel h is equal to the Fourier transform of convolution $g * h$

$$\left|G(u,v)H(u,v)\right| =$$

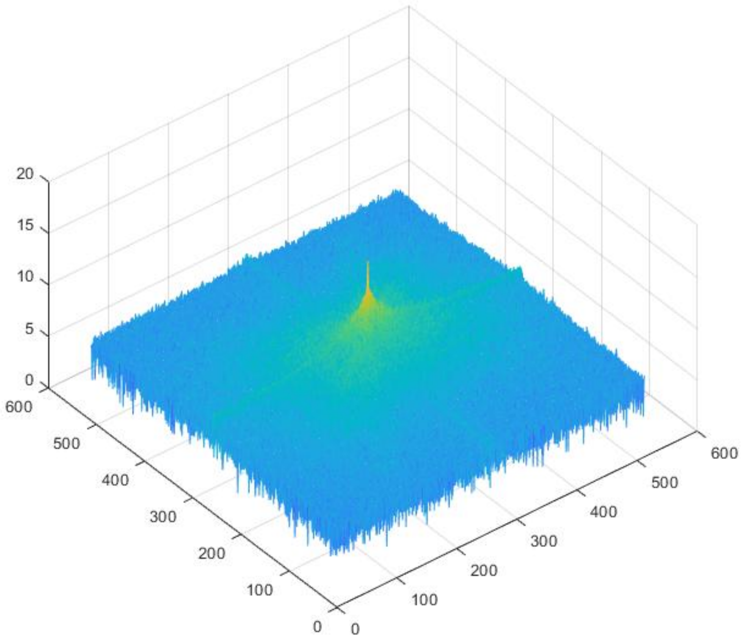


$$\otimes$$

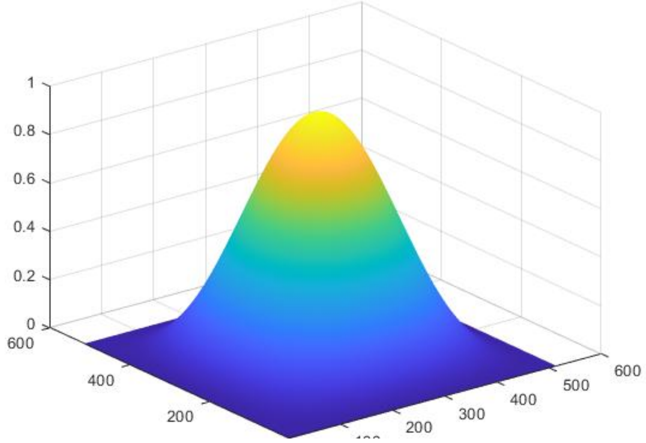


Frequency Domain Filtering

$$\left|G(u,v)H(u,v)\right| =$$

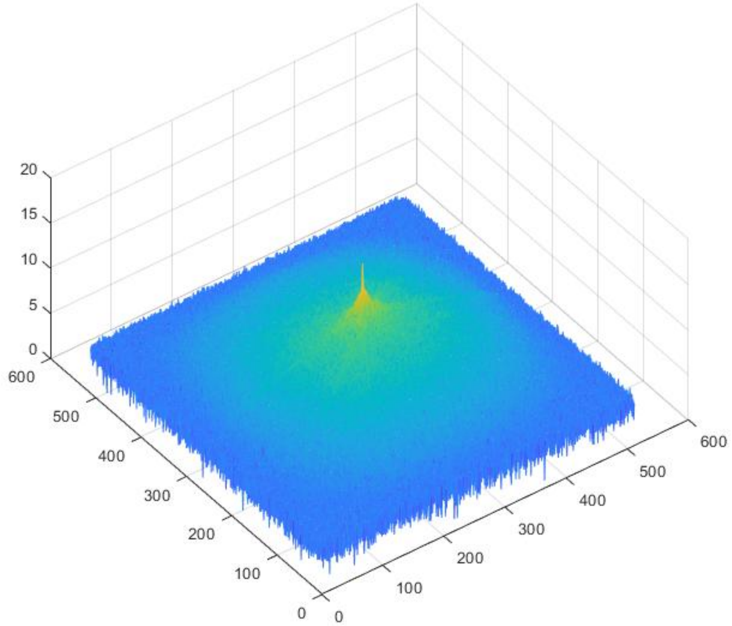


$\otimes$



$=$

$=$



Frequency Domain Filtering

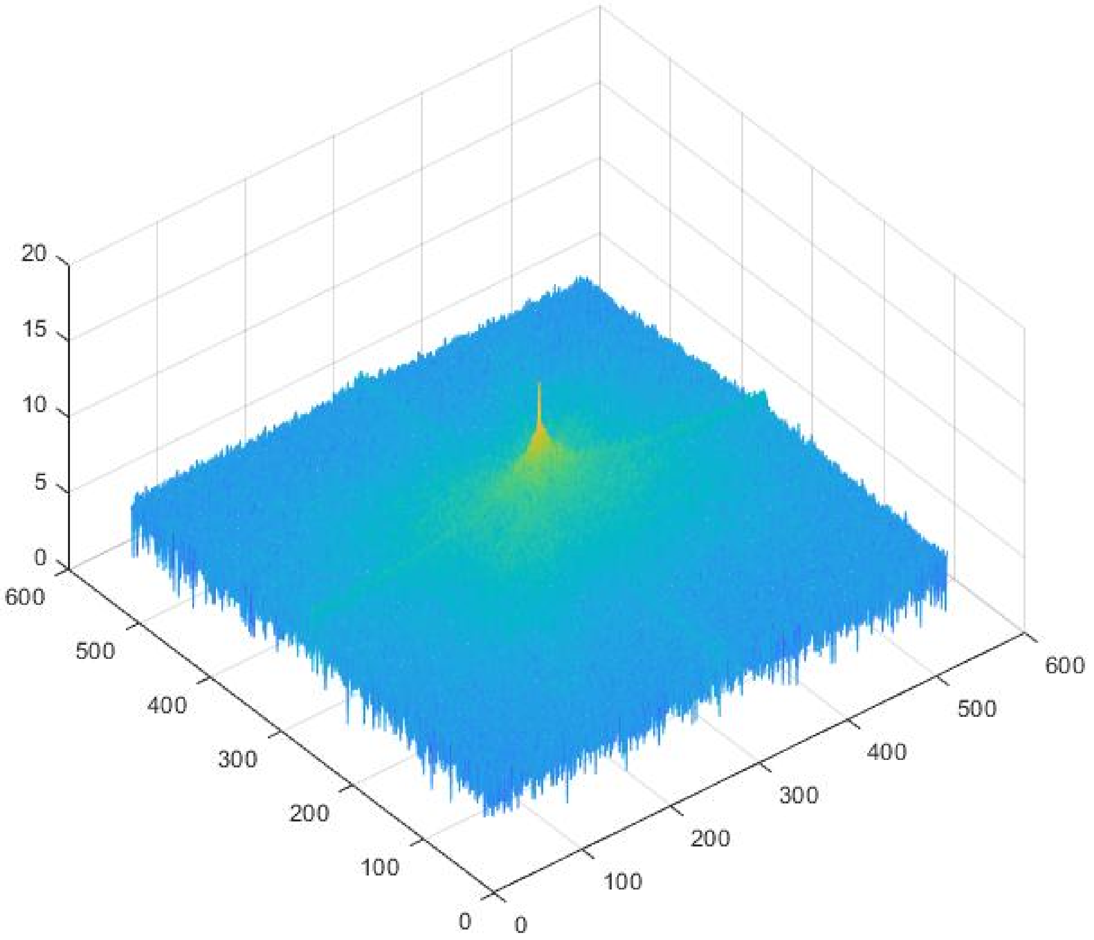
- According to the Convolution Theorem Inverse Fourier Transform from $G(u,v)H(u,v)$ returns a filtered image $F^{-1}\left(\underbrace{G(u,v)}_{F(g(x,y))}H(u,v)\right)$

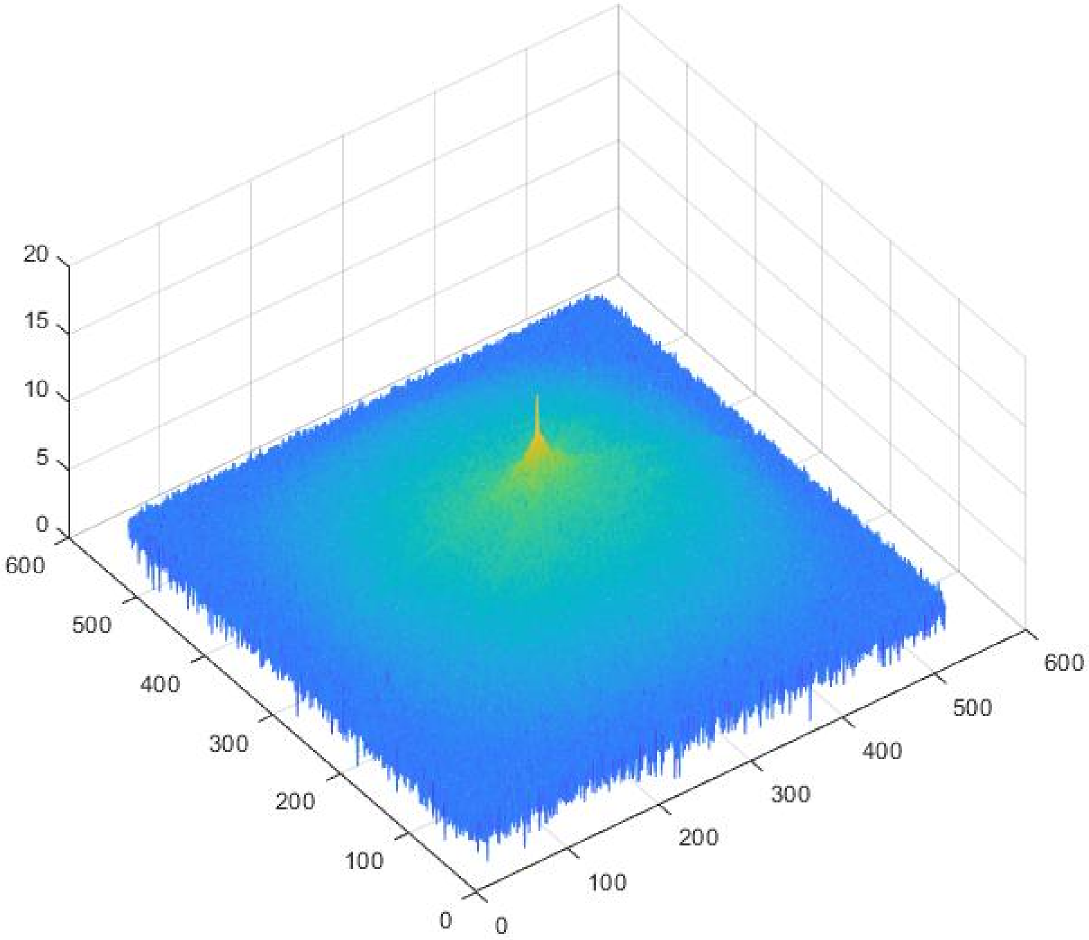
Frequency Domain Filtering

“Lena” corrupted with additive Gaussian noise 0.2σ (image g)

The result of filtering (image $\hat{f}$)

Frequency Domain Filtering





Fourier Power spectrum $|G|$ of noisy "Lena"   Fourier power spectrum $|G(u,v)H(u,v)| = |\hat{F}(u,v)|$ of filtered "Lena"

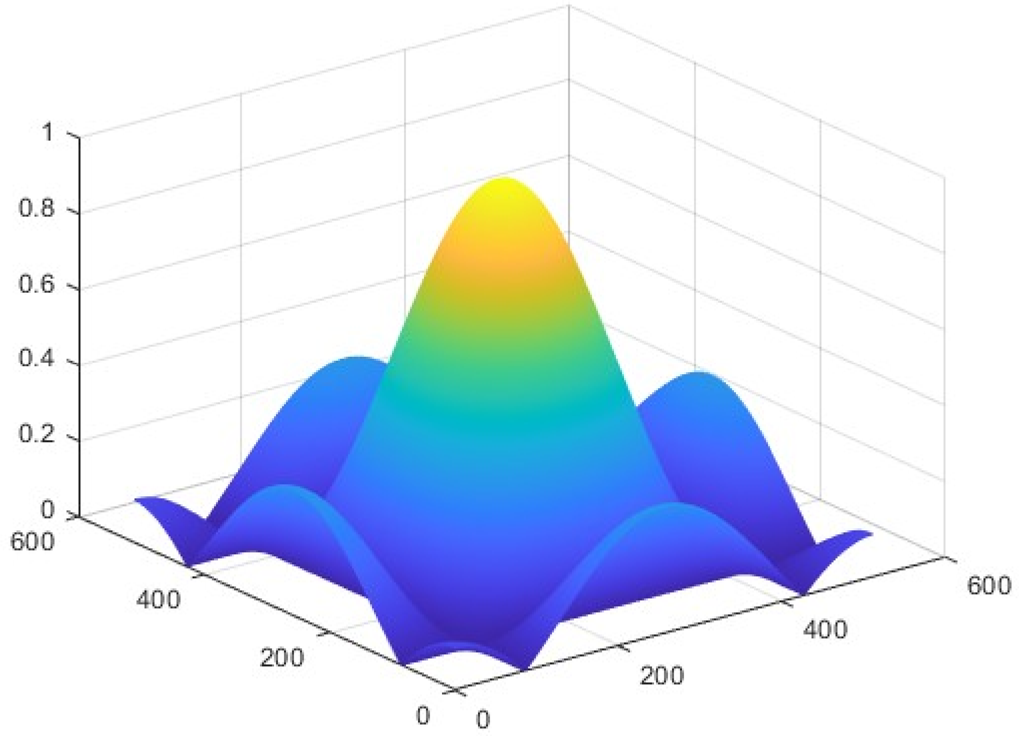
Frequency Domain Representation of Popular Linear Filters

Fourier Power Spectra of Popular Filter Kernels

Low Pass Filter - Mean Filter 1

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

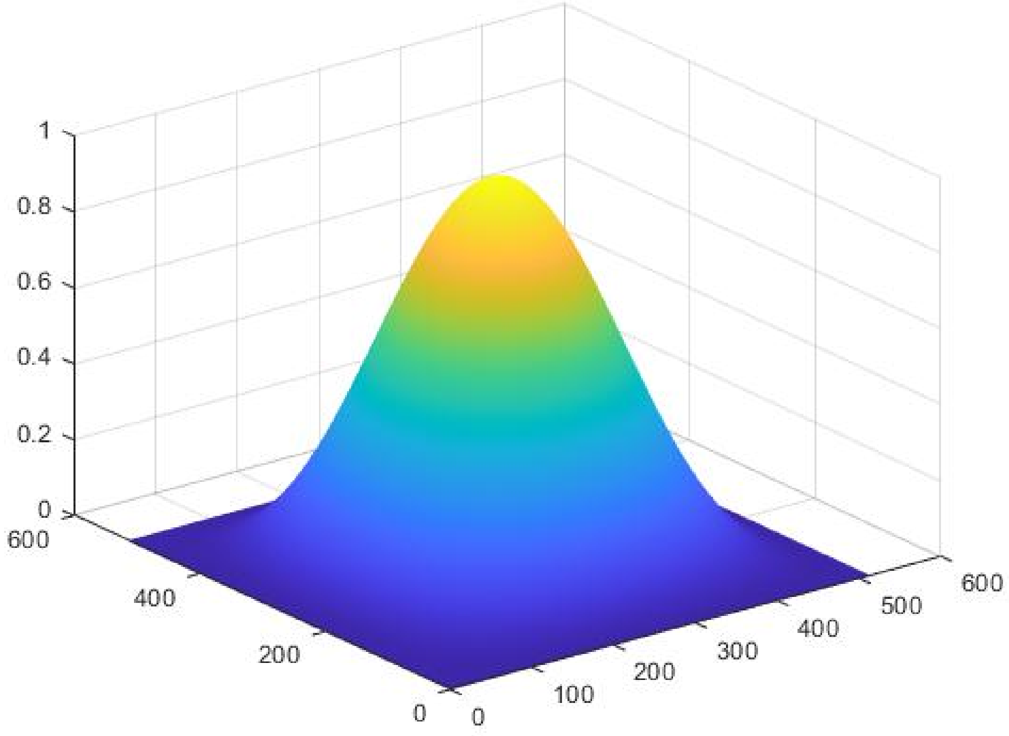
$$W = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



Low Pass Filter - Smart Filter 1

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

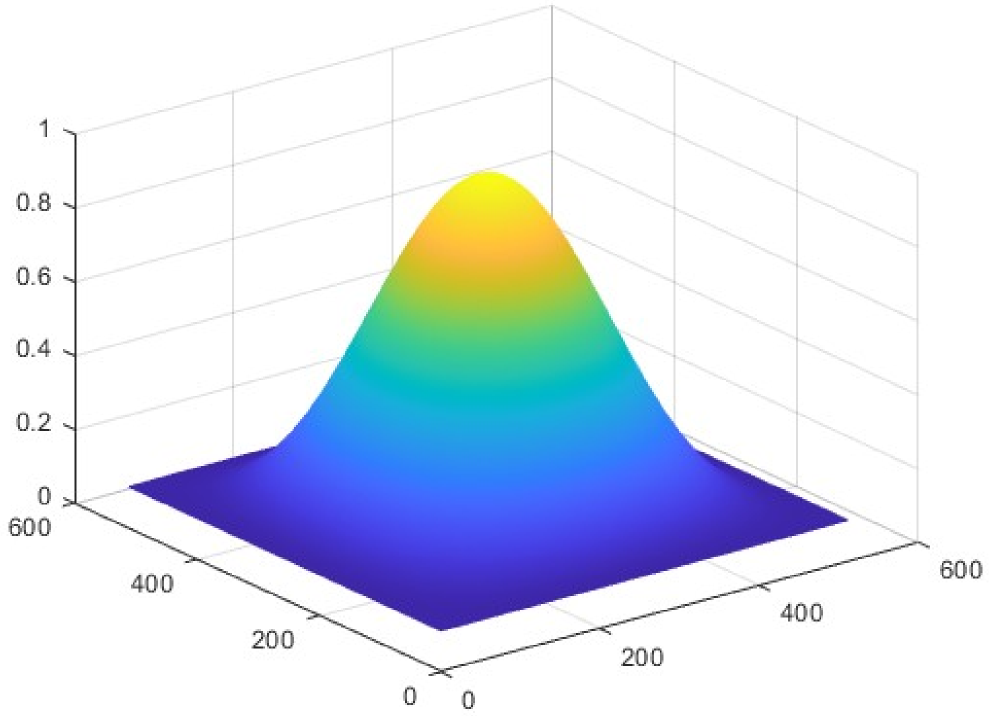
$$W = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$



Low Pass Filter - Smart Filter 2

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

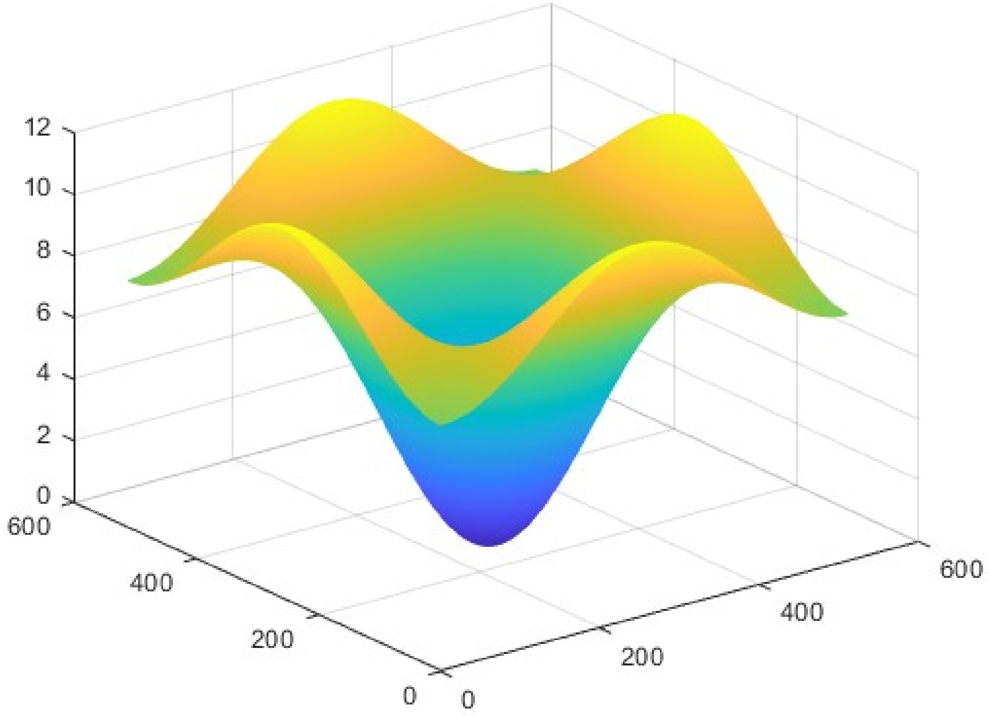
$$W = \frac{1}{18}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 6 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$



High Pass Filter –
Laplace 2 Downward Edge Detection

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

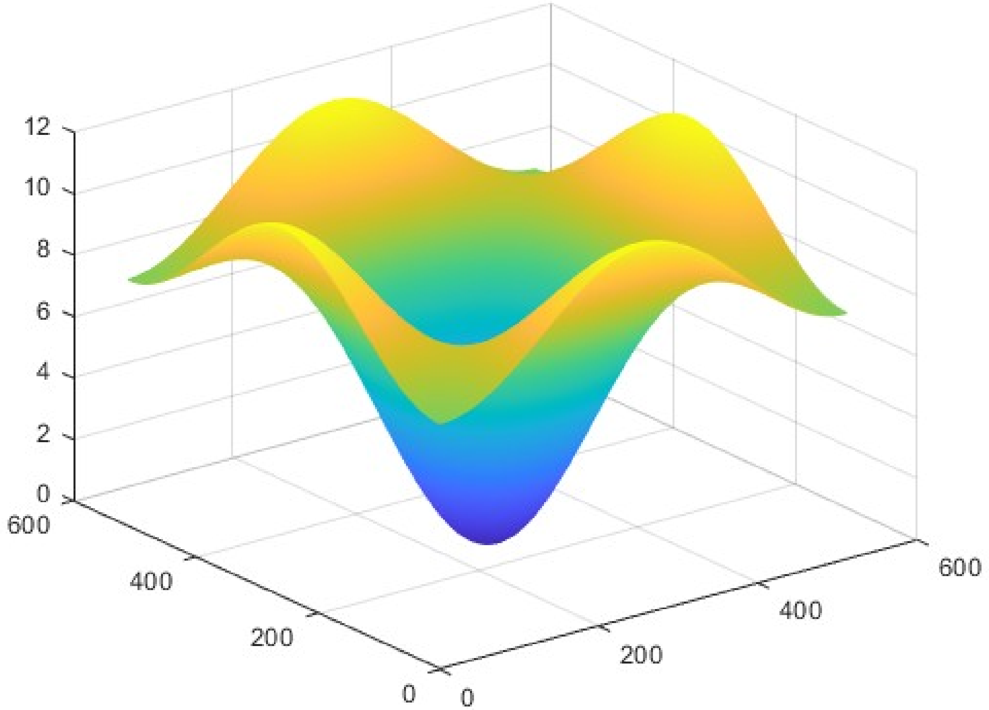
$$W = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



High Pass Filter – Laplace 2 Upward Edge Detection

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

$$W = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

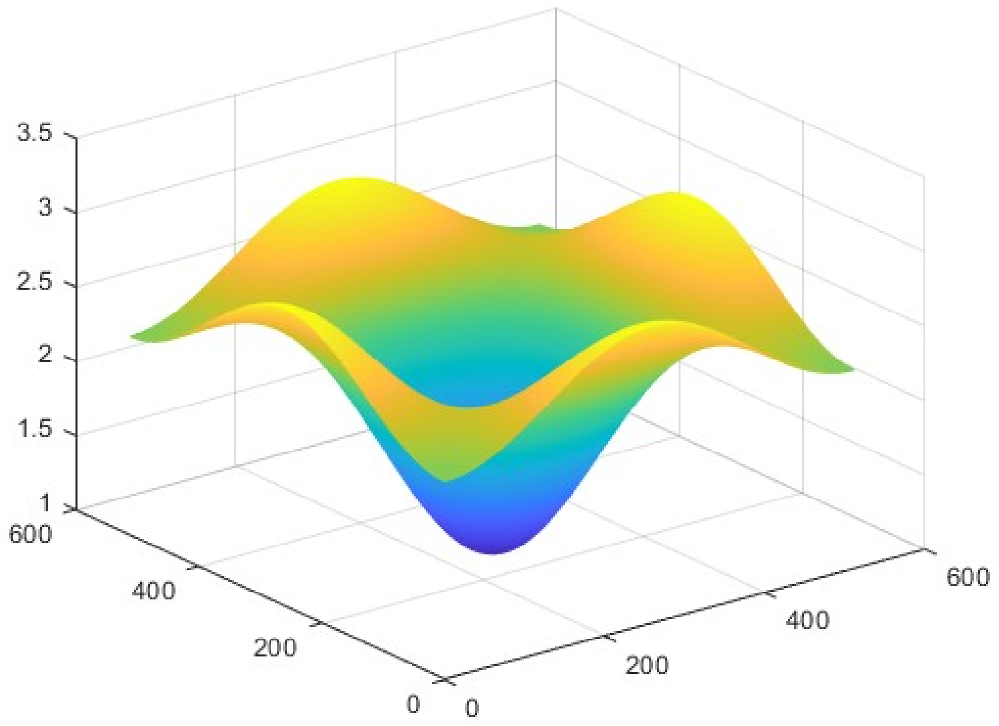


Unsharp Masking Filter – High Frequency Enhancement

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

$$\begin{pmatrix} -\frac{k}{9} & -\frac{k}{9} & -\frac{k}{9} \\ -\frac{k}{9} & k+1-\frac{k}{9} & -\frac{k}{9} \\ -\frac{k}{9} & -\frac{k}{9} & -\frac{k}{9} \end{pmatrix}$$

$$k = 1.5$$

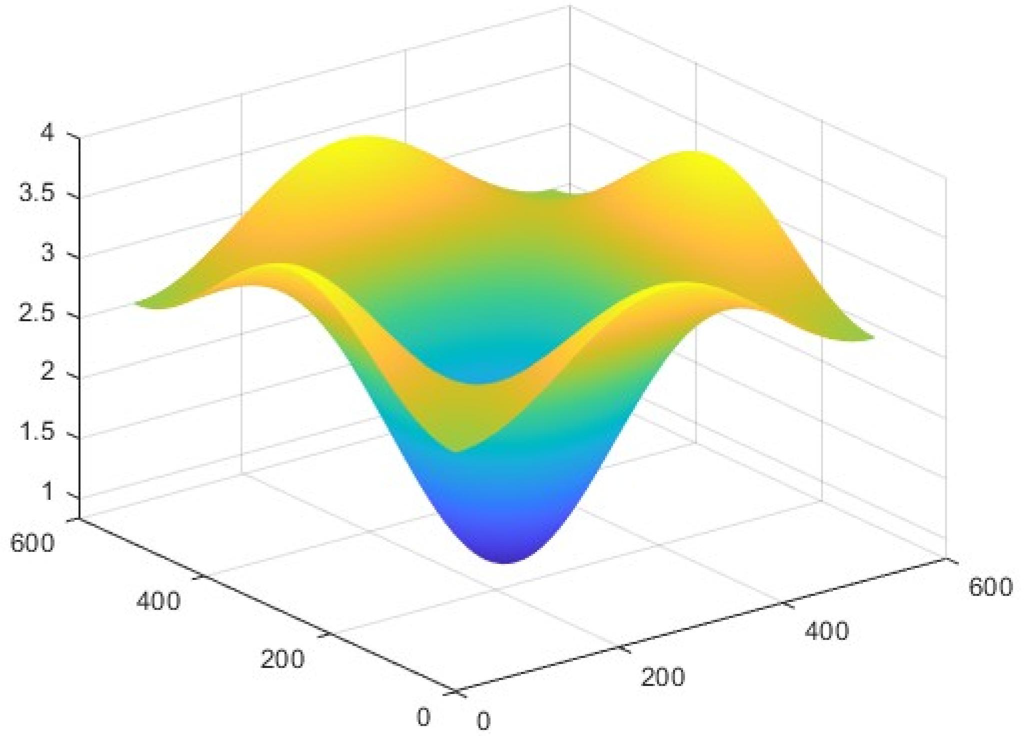


Unsharp Masking Filter – High Frequency Enhancement

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

$$\begin{pmatrix} -\frac{k}{9} & -\frac{k}{9} & -\frac{k}{9} \\ -\frac{k}{9} & k+1-\frac{k}{9} & -\frac{k}{9} \\ -\frac{k}{9} & -\frac{k}{9} & -\frac{k}{9} \end{pmatrix}$$

$k = 2.0$

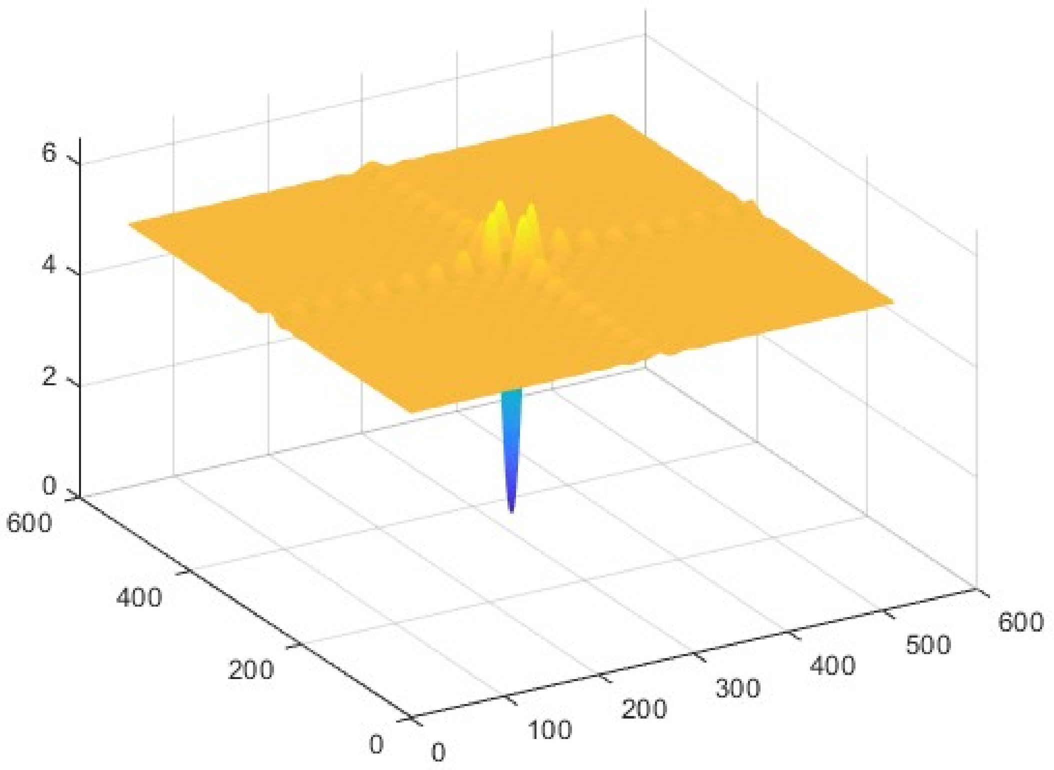


Global Frequency Correction–
High and Medium Frequency Enhancement

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

$$\begin{pmatrix} \frac{k_3-k_2}{mn} & \cdots & \frac{k_3-k_2}{mn} & \cdots & \frac{k_3-k_2}{mn} \\ \cdots & & \cdots & & \cdots \\ \frac{k_3-k_2}{mn} & \cdots & k_1+k_2+\frac{k_3-k_2}{mn} & \cdots & \frac{k_3-k_2}{mn} \\ \cdots & & \cdots & & \cdots \\ \frac{k_3-k_2}{mn} & \cdots & \frac{k_3-k_2}{mn} & \cdots & \frac{k_3-k_2}{mn} \end{pmatrix}$$

$m=n=35; k_1=k_3=0.5; k_2=5.0$

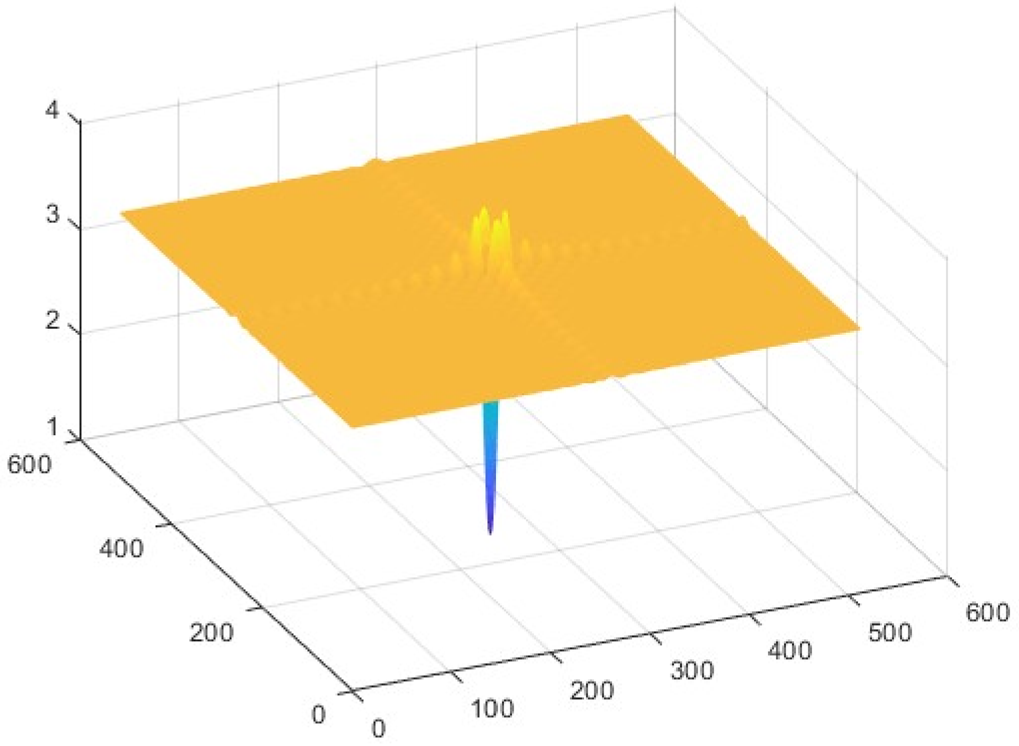


Global Frequency Correction–
High and Medium Frequency Enhancement

- This is a power Fourier spectrum $|\boldsymbol{H}|$ (Fourier Transform magnitude) of 512x512 $\boldsymbol{h}$ corresponding to a linear filter:

$$\begin{pmatrix} \frac{k_3-k_2}{mn} & \dots & \frac{k_3-k_2}{mn} & \dots & \frac{k_3-k_2}{mn} \\ \dots & & \dots & & \dots \\ \frac{k_3-k_2}{mn} & \dots & k_1+k_2+\frac{k_3-k_2}{mn} & \dots & \frac{k_3-k_2}{mn} \\ \dots & & \dots & & \dots \\ \frac{k_3-k_2}{mn} & \dots & \frac{k_3-k_2}{mn} & \dots & \frac{k_3-k_2}{mn} \end{pmatrix}$$

$$m = n = 51; k_1 = k_3 = 0.5; k_2 = 3.0$$



Periodic and Quasi-Periodic Noise Filtering

Periodic Noise

- Periodic Noise is a sinusoidal wave with the frequency r (period $1/r$) added to a signal
- In the spatial domain, this noise corrupts the entire signal
- In the **frequency domain**, it corrupts only a few spectral coefficients corresponding to the frequency of the noisy wave. This kind of noise leads to unusually high magnitude of the corresponding spectral coefficients
- Thus, in the frequency domain, this noise can be reduced or even completely removed if the corresponding spectral coefficients have been corrected

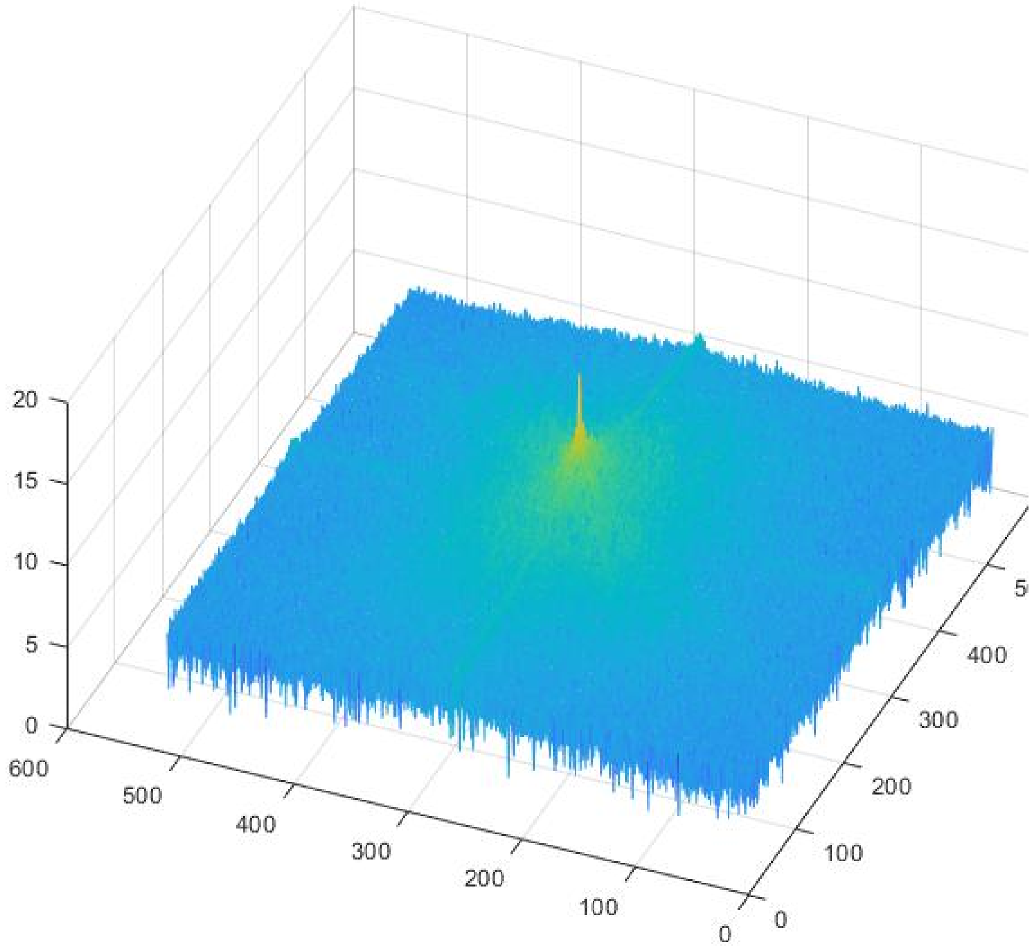
Periodic Noise

Original Image

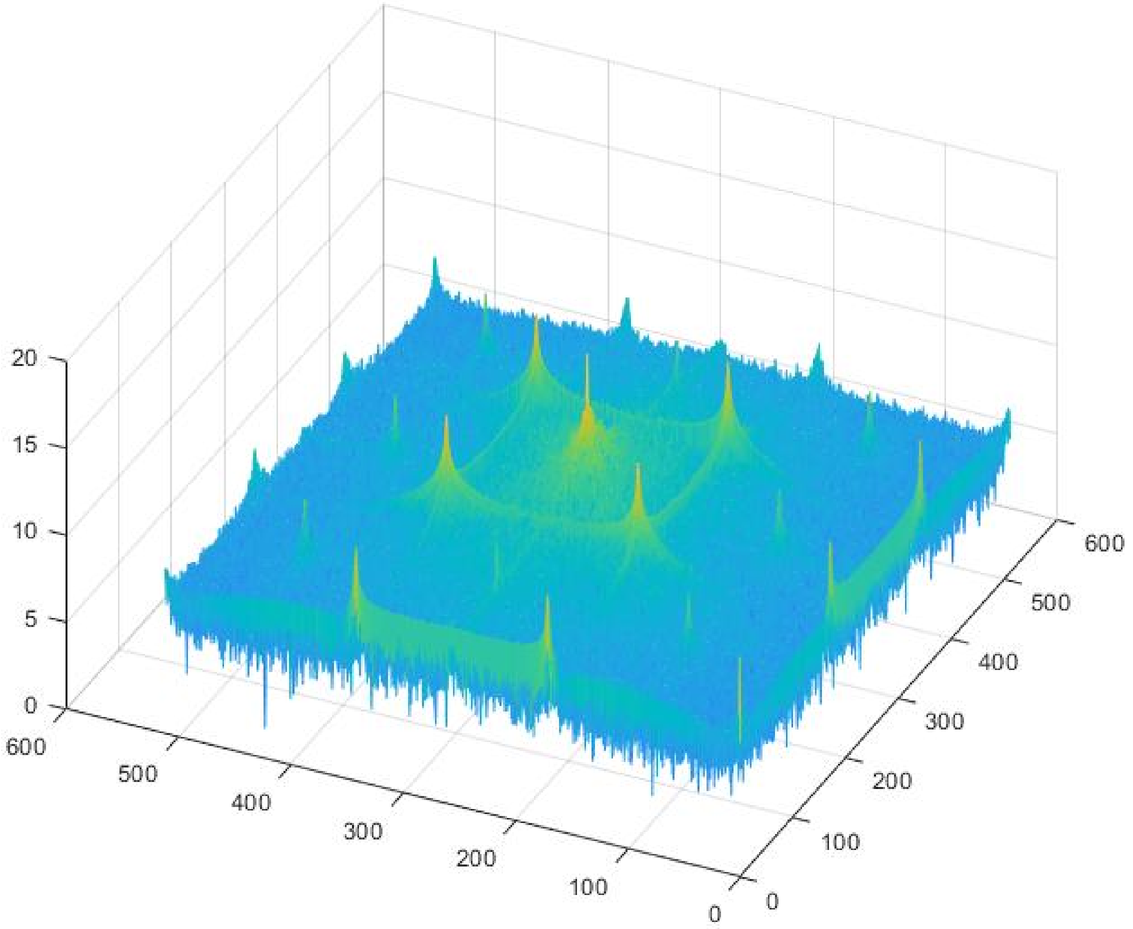
Image corrupted by periodic noise (moire)

Periodic Noise

Periodic noise (moire) is resulted in abnormal peaks in the magnitudes corresponding to the frequencies of moire patterns



Fourier Power Spectrum of the Original Image



Fourier power spectrum of the Image corrupted by periodic noise (moire)

Moire Effect

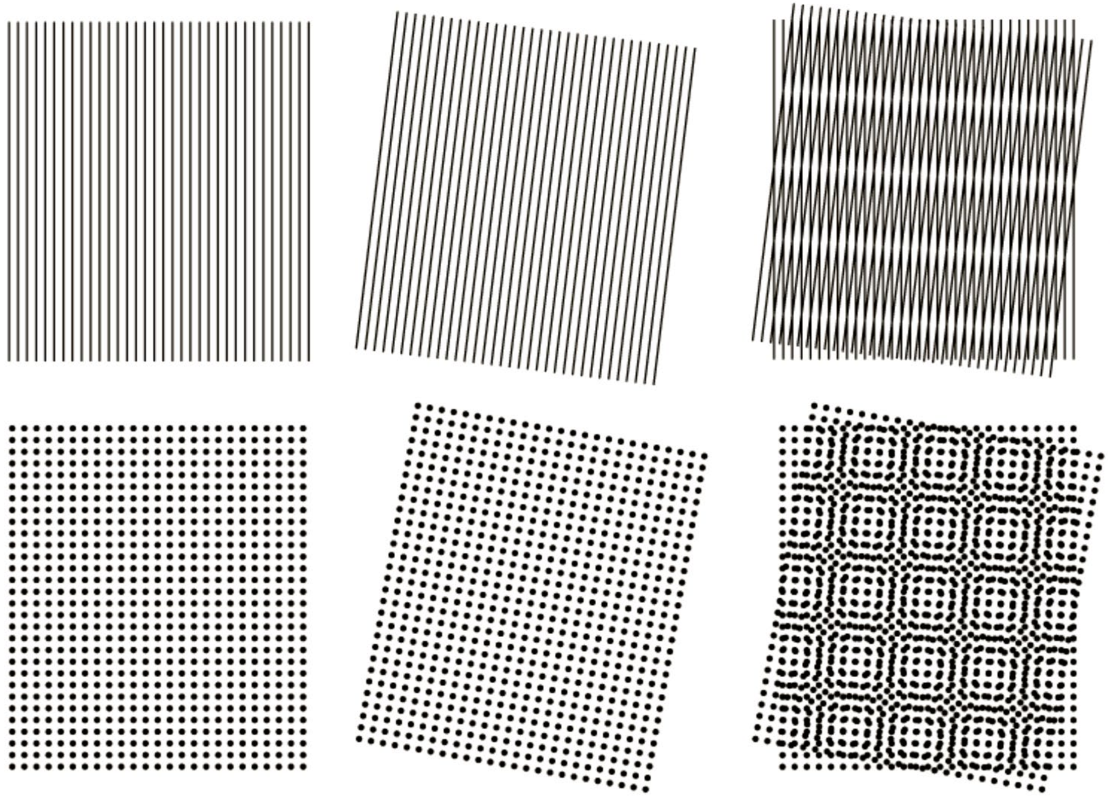


FIGURE 4.20 Examples of the moiré effect. These are ink drawings, not digitized patterns. Superimposing one pattern on the other is equivalent mathematically to multiplying the patterns.

Moire Effect

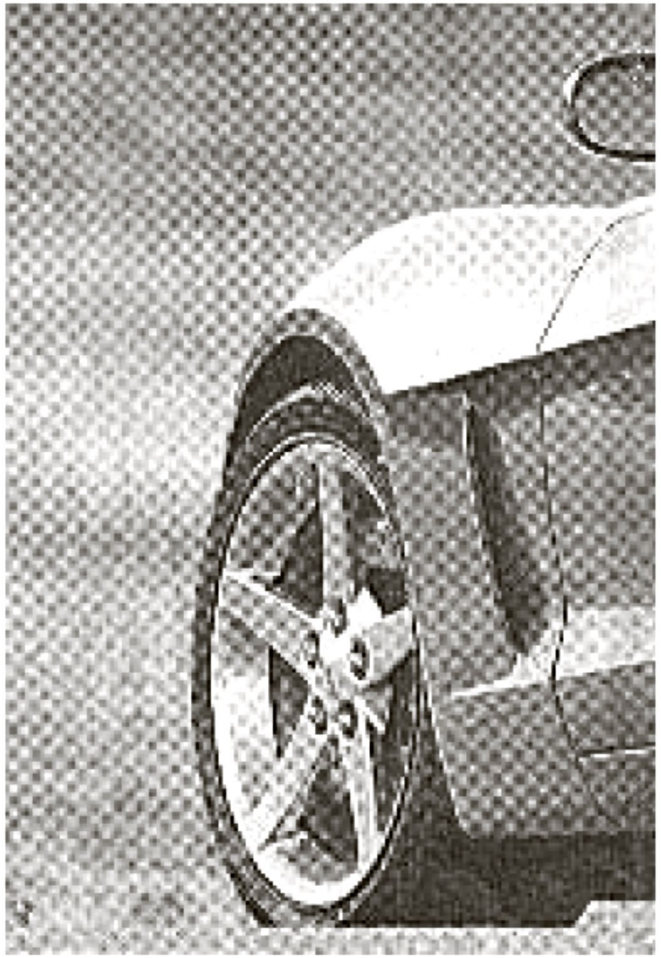


FIGURE 4.21 A newspaper image of size 246×168 pixels sampled at 75 dpi showing a moiré pattern. The moiré pattern in this image is the interference pattern created between the $\pm 45°$ orientation of the halftone dots and the north–south orientation of the sampling grid used to digitize the image.

Moire Effect

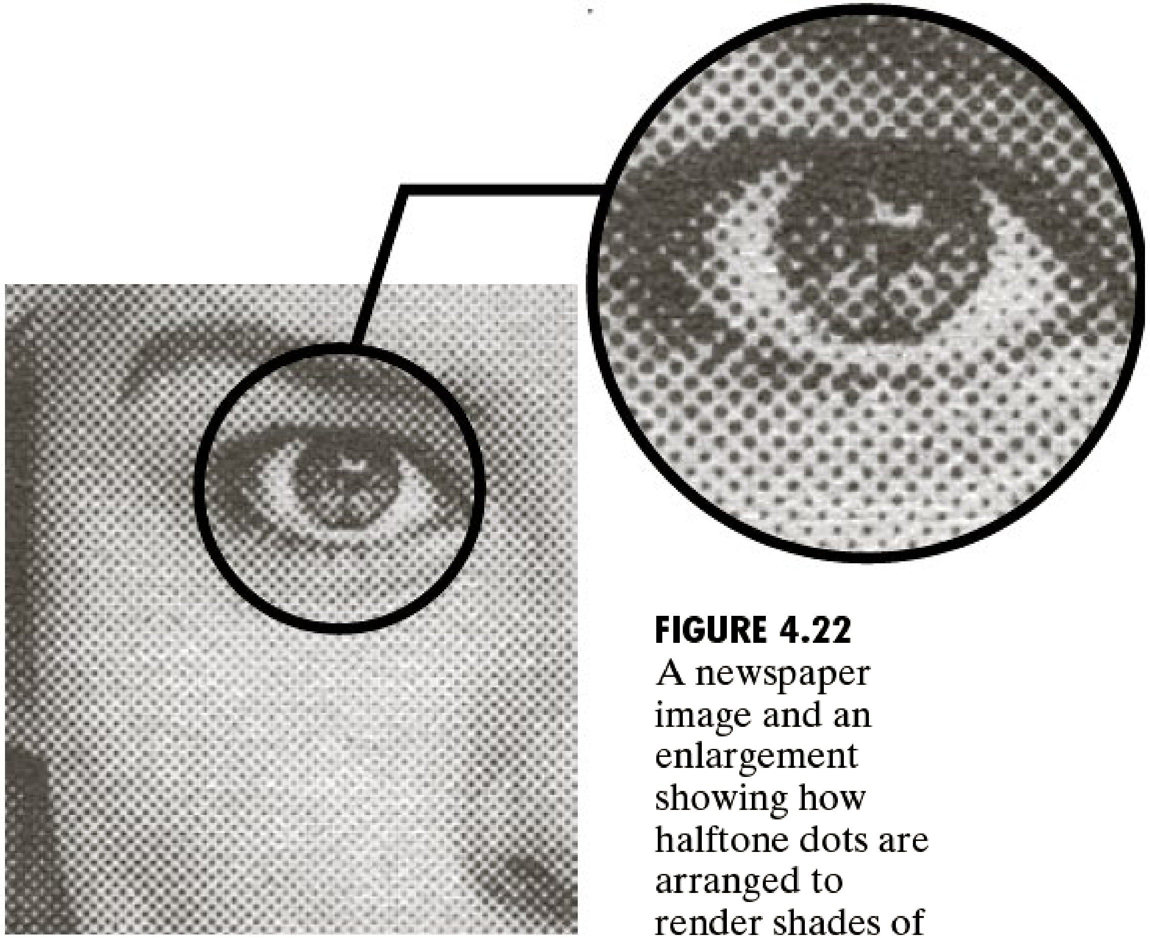
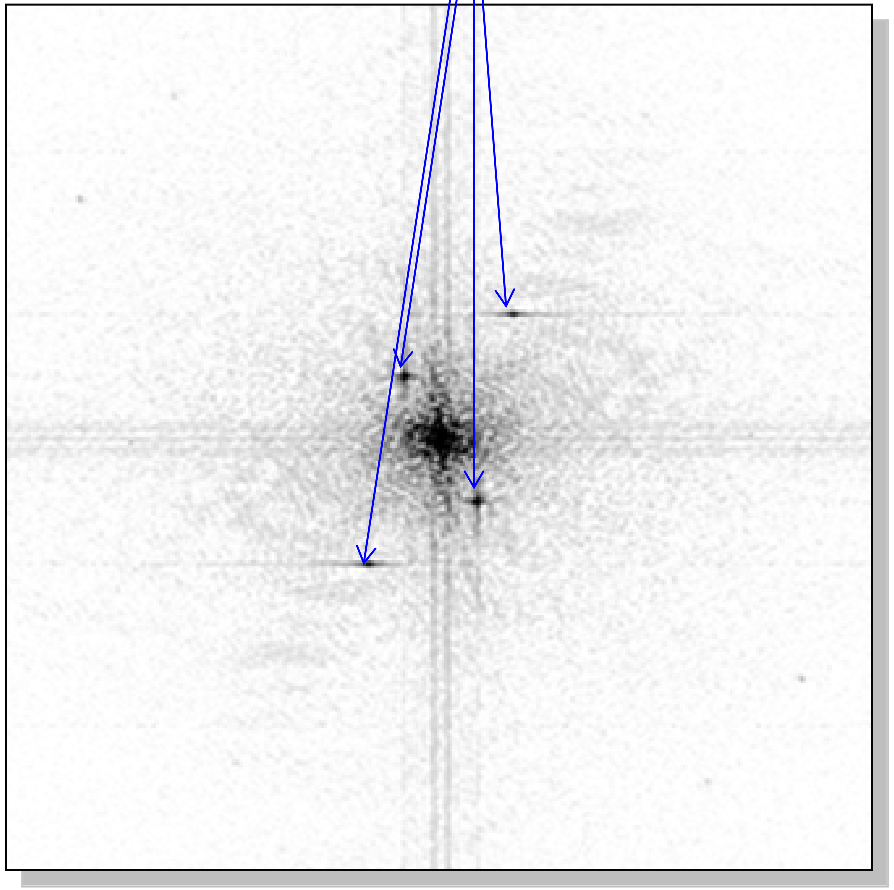


FIGURE 4.22 A newspaper image and an enlargement showing how halftone dots are arranged to render shades of gray.

Image

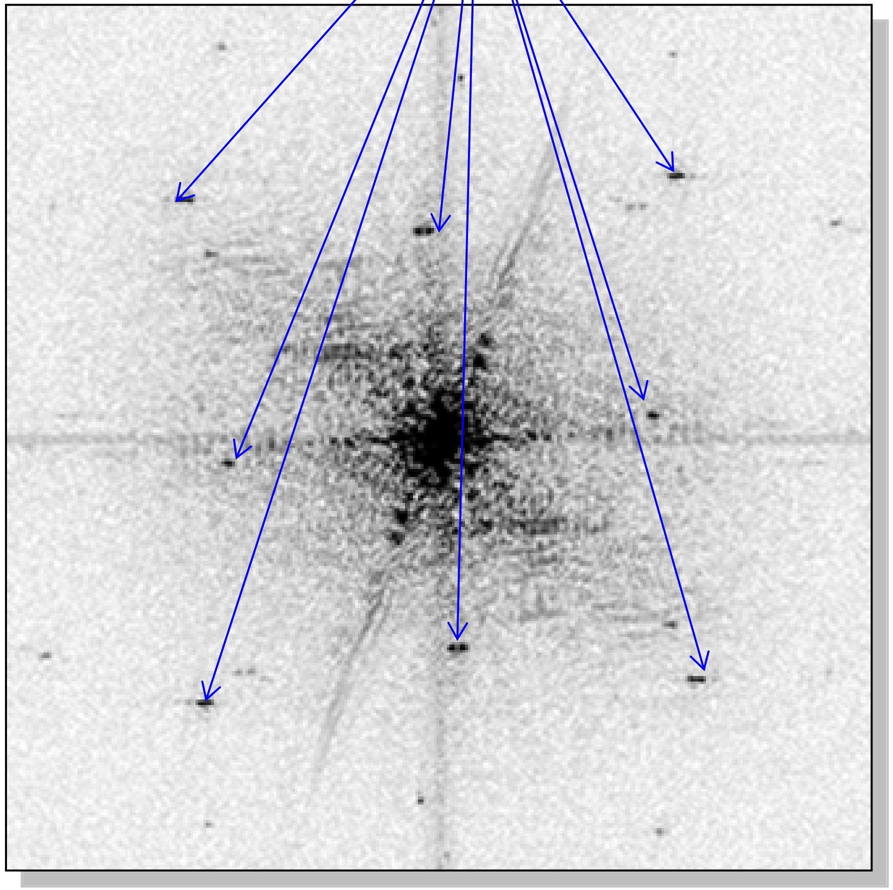
Corrupted magnitudes of the spectral coefficients corresponding to the noise frequencies



Its power spectrum

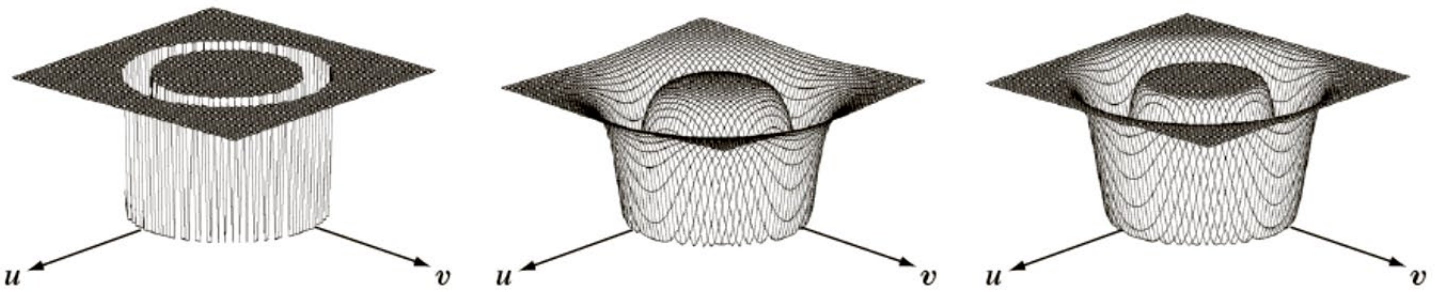
Image

Corrupted magnitudes of the spectral coefficients corresponding to the noise frequencies



Its power spectrum

Types of Frequency Domain Filters



a b c

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

TABLE 4.4
Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

“Ideal” Mask Generation

- Let $D(u, v) = |G(u, v)|$- magnitude of the Fourier transform coefficient corresponding to the frequencies u, v. Then

$$M(u,v) = \begin{cases} 1 & , \text{if } D(u,v) \le T \\ 0 & , \text{if } D(u,v) > T \end{cases}$$

Moreover if $D(u,v)$ is larger than T for some (u,v), we not just put one 0 in the mask, we put a circle of zeroes with center at (u,v) to improve the result.

- To filter, we implement component-wise multiplication of D and M:

$$\tilde{D} = D \otimes M$$

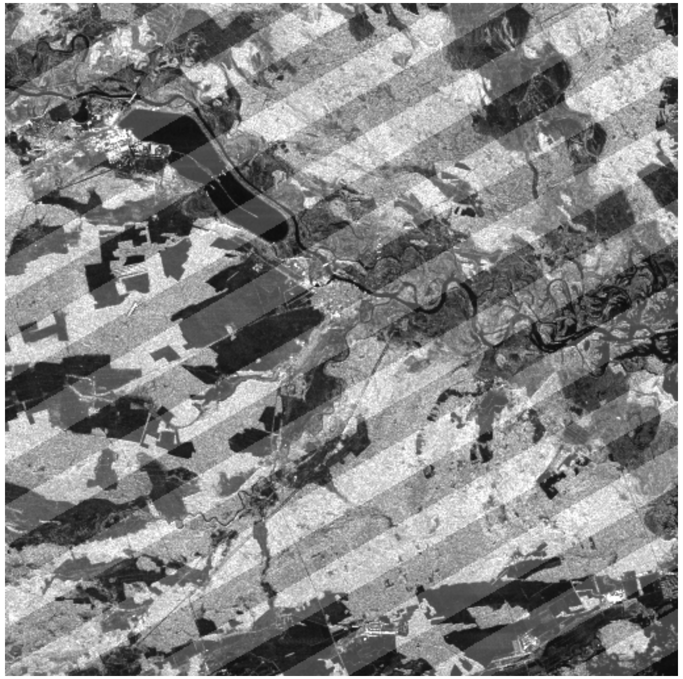
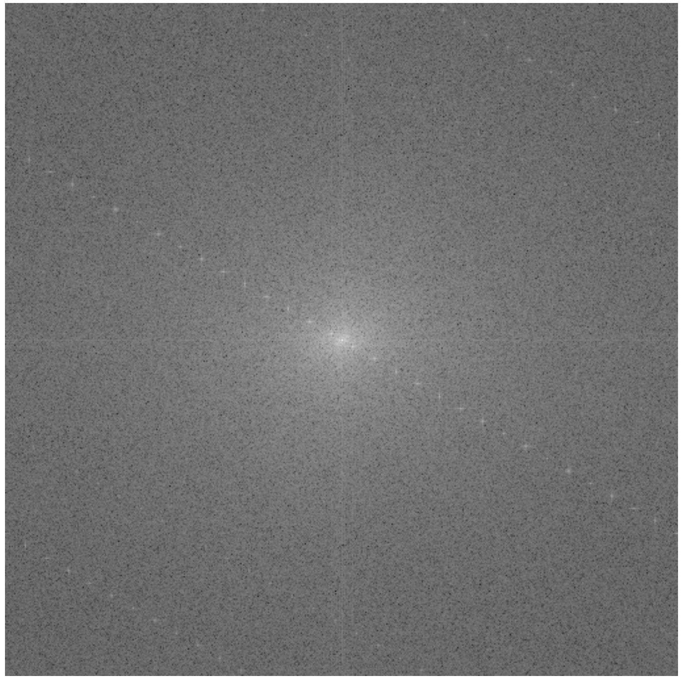
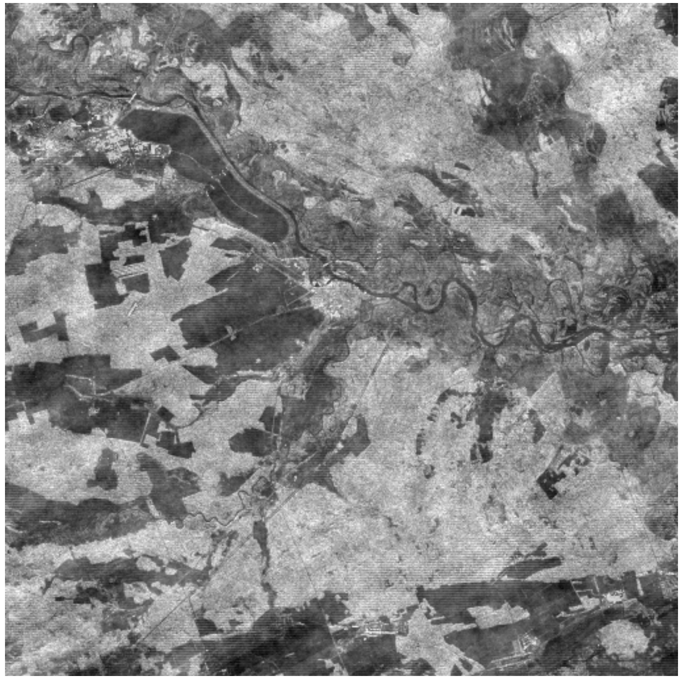


Image with periodic stripes

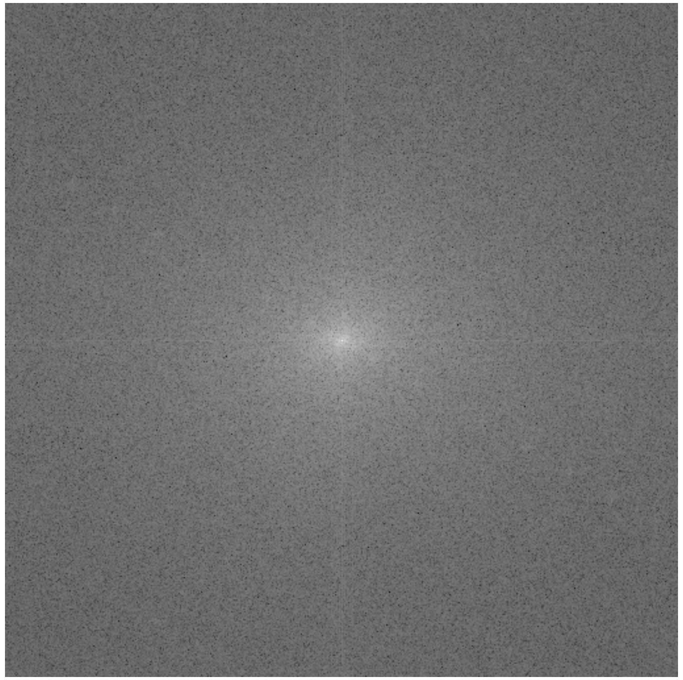
Its power spectrum





Filtered using “ideal” mask, T=1.5

Power spectrum of the filtered image



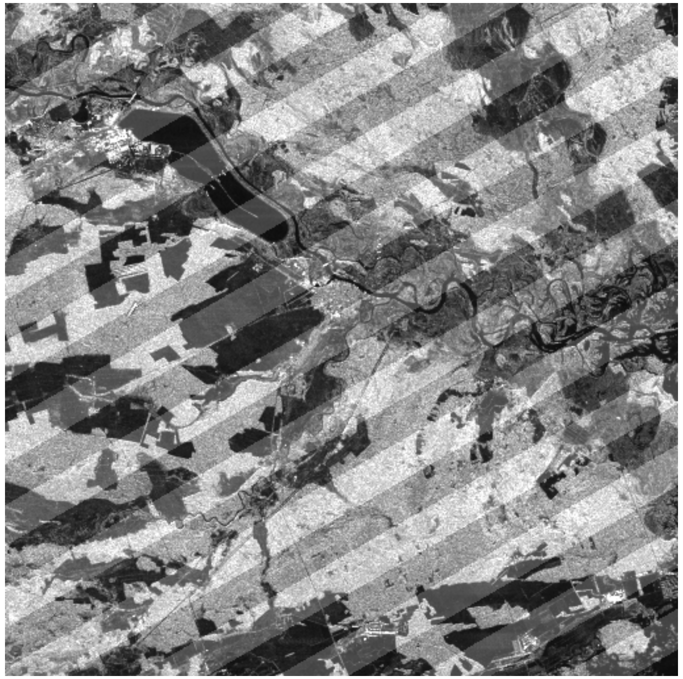
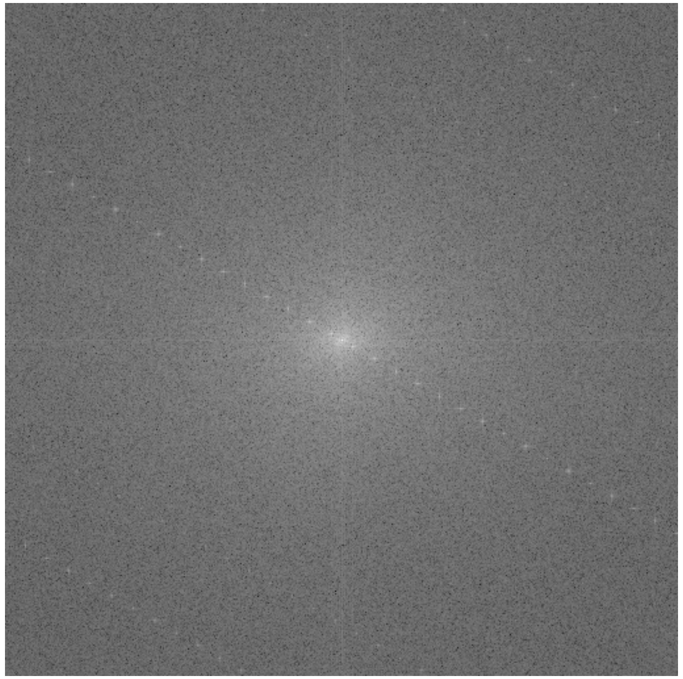
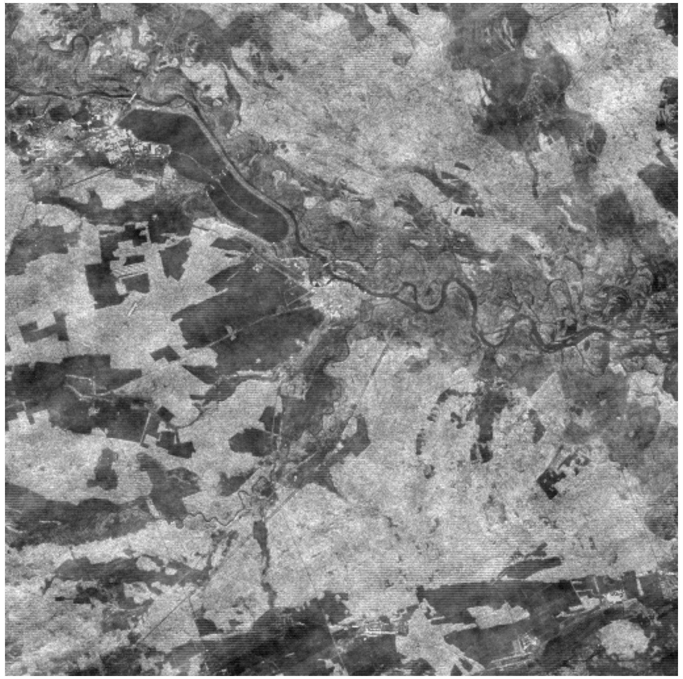


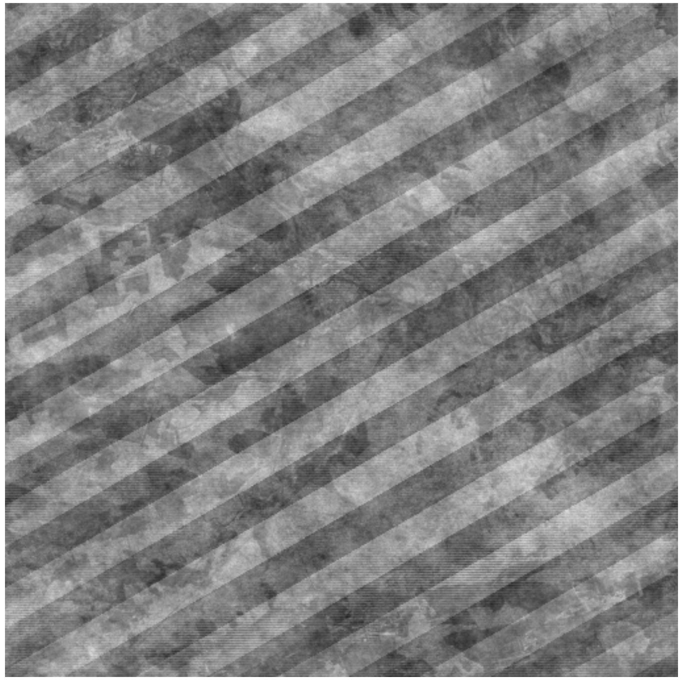
Image with periodic stripes



Its power spectrum



Filtered using "ideal" mask, *T*=1.5

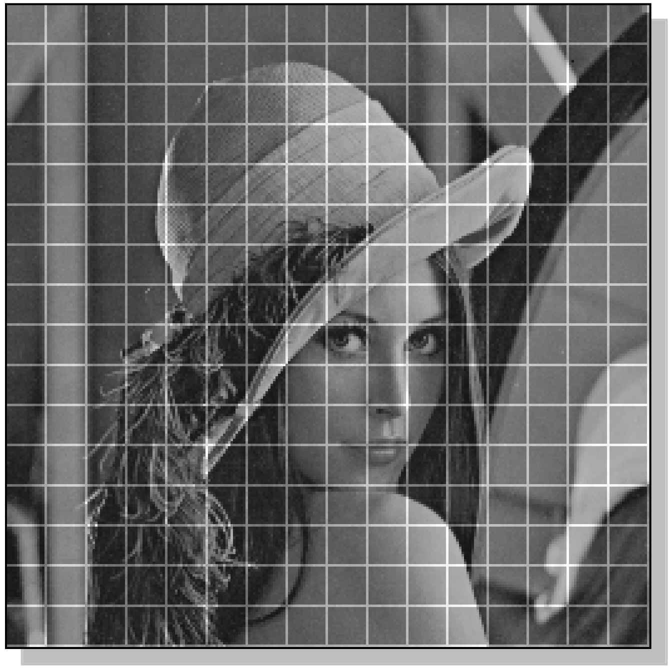


Difference between noisy and filtered images

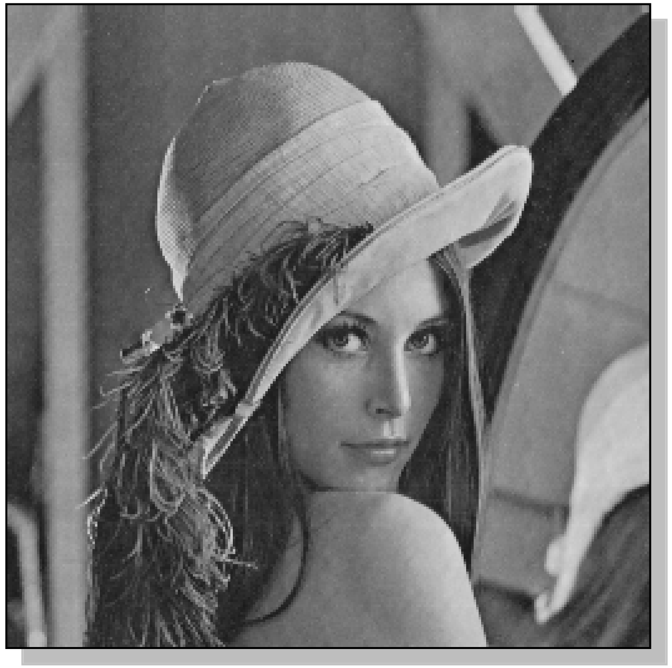
Mean Filter in Frequency Domain

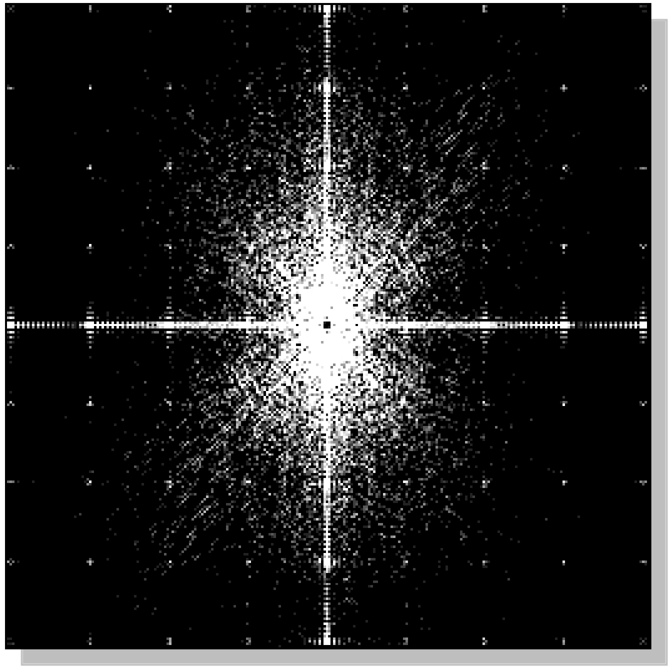
$$\tilde{D}(u,v) = \begin{cases} D(u,v) & , \text{if } \dfrac{D(u,v)}{S(u,v)} \le T \\ D(u,v)/d & , \text{otherwise} \end{cases}$$

- **This filter and the next filters are utilized using local n x m neighborhood processing in the frequency domain (instead of convolution with a fixed kernel)**
- Therefore, this and the next filters are adaptive
- $S(u,v)$ is the local mean in the moving window around the coefficient $D(u,v)$.
- T is the threshold
- $\tilde{D}(u,v)$ is the filtered value of $D(u,v)$
- d - is the parameter specifying the strength of peak reduction

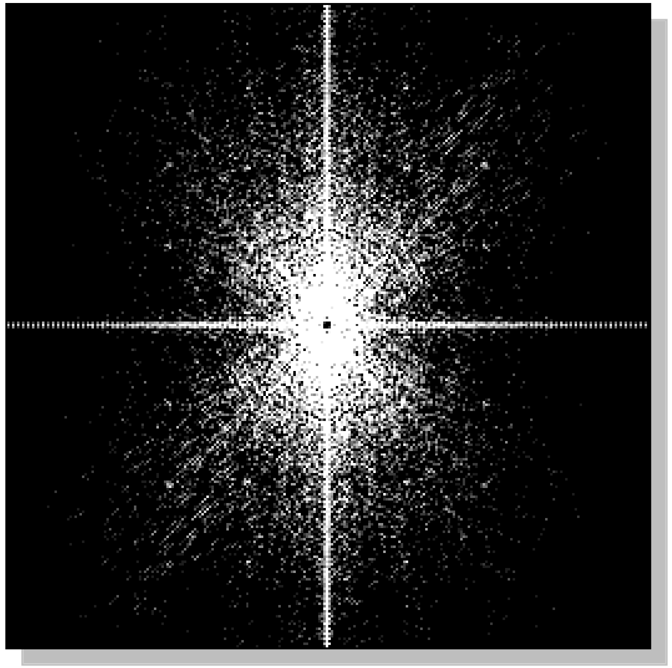


Mean Spectral Filter With T=5, d=50, 3x3 window

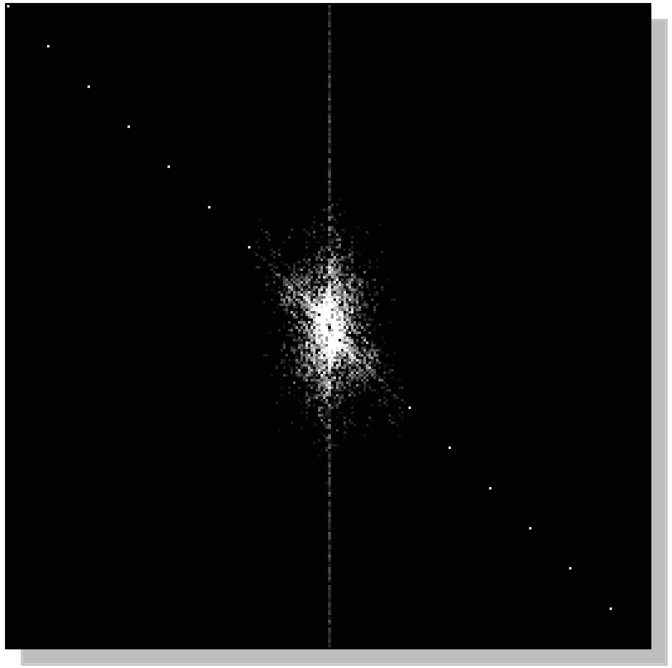




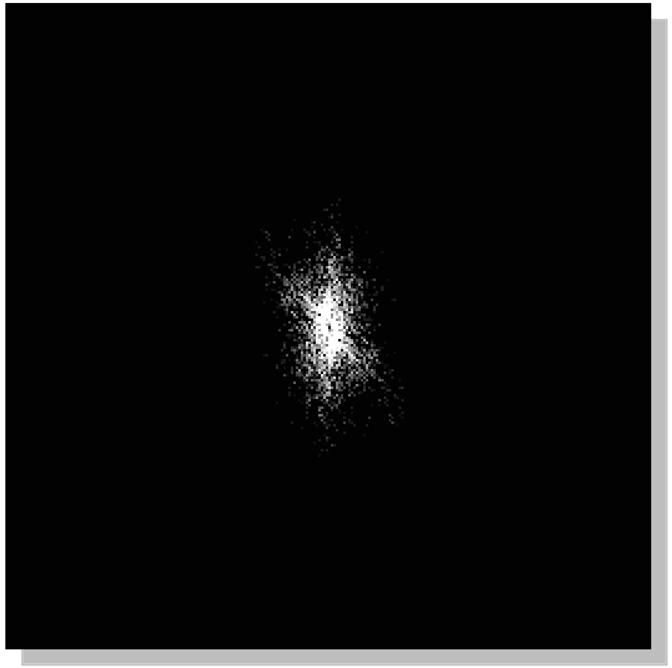
Enhanced Power Spectrum



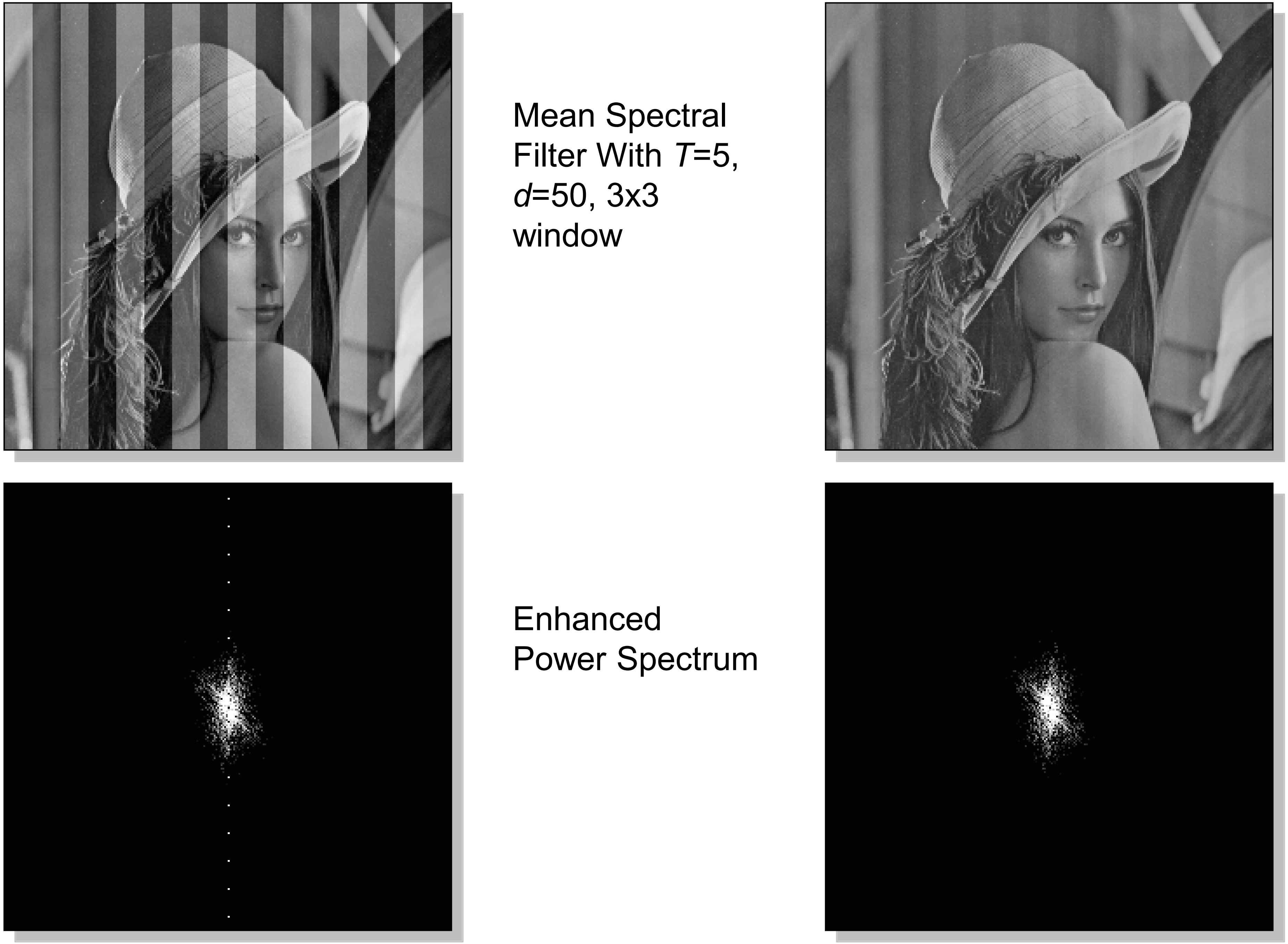
Mean Spectral Filter With *T*=5, *d*=50, 3x3 window



Enhanced Power Spectrum



Mean Spectral Filter With T=5, d=50, 3x3 window



Enhanced Power Spectrum

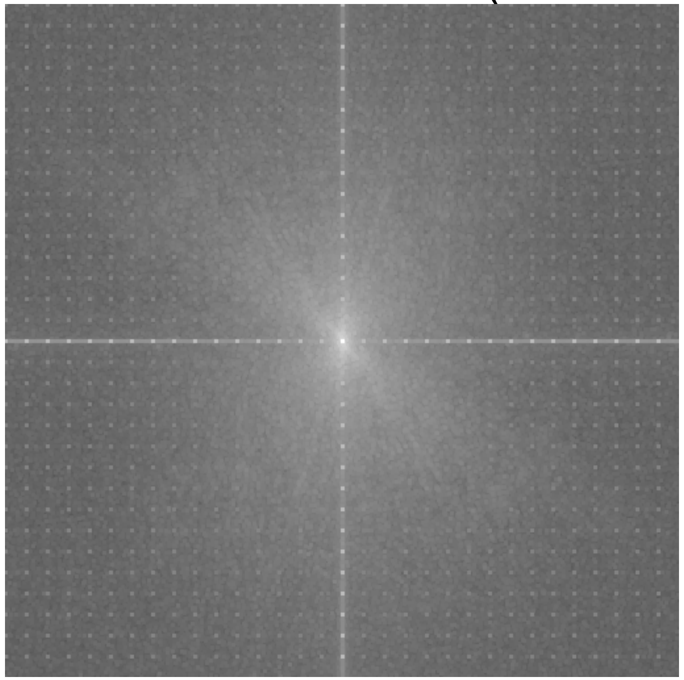
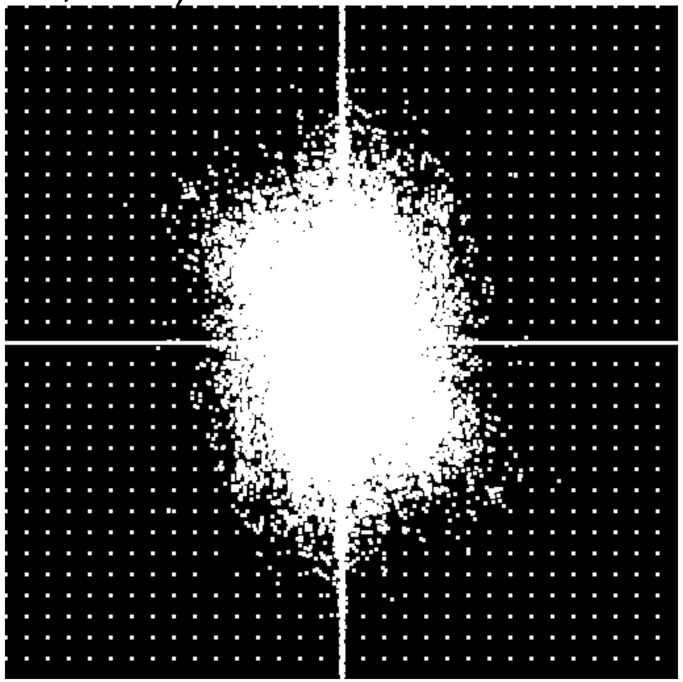
Median Based Peak Detector

$D(u,v)$ is a peak if:

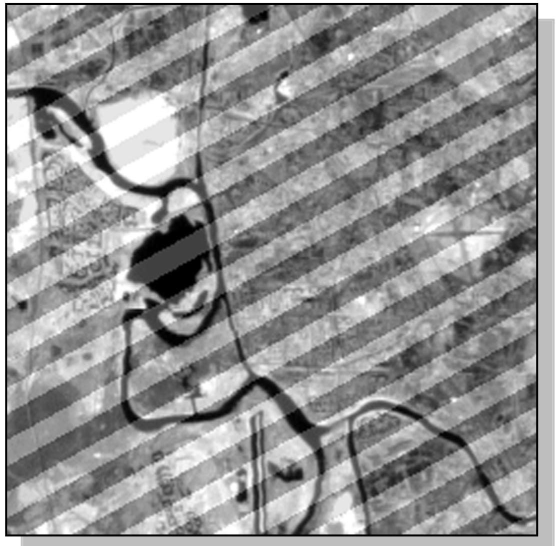
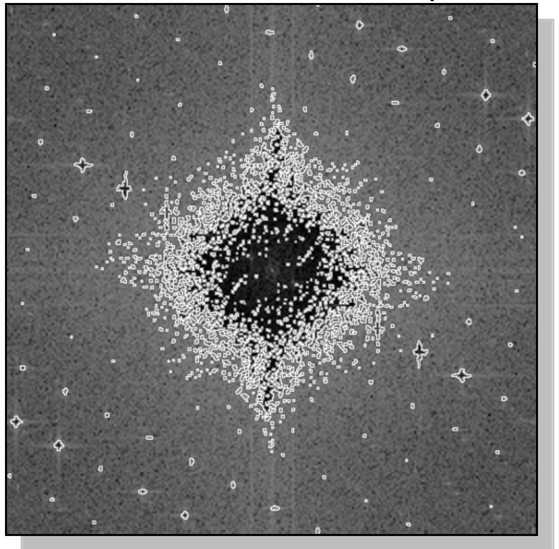
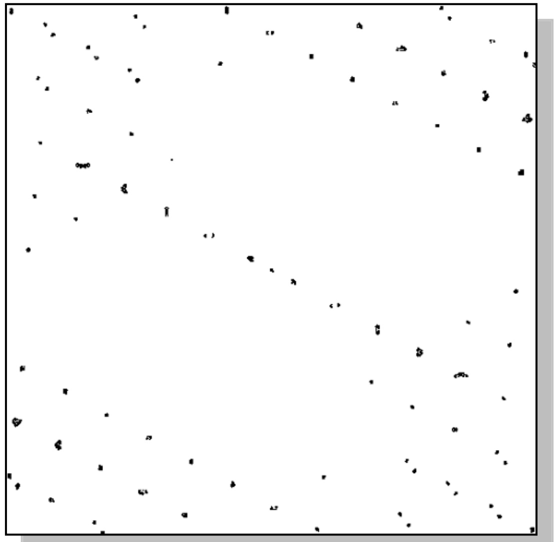
$$\frac{D(u,v)}{MED\{D(u,v)\}} > T$$

Image | Enhanced Power Spectrum | Detected Peaks

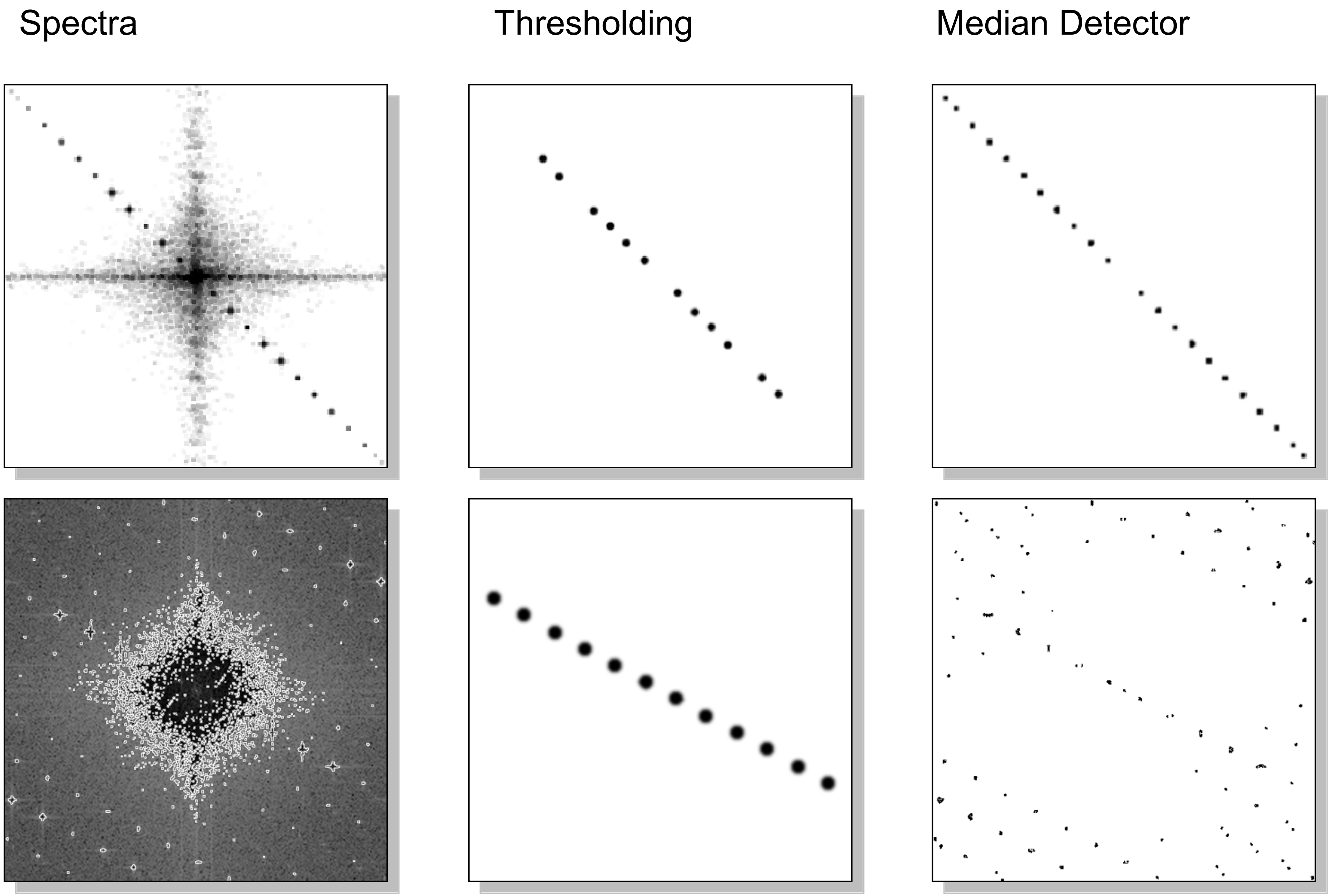
(11x11 window, T=8)

(11x11 window, T=6)

Comparison With Thresholding



Spectral Median Filter

The peak is substituted by median:

$$\tilde{D}(u,v) = \begin{cases} MED\{D(u,v)\} & \text{, if } \dfrac{D(u,v)}{MED\{D(u,v)\}} > T \\ D(u,v) & \text{, otherwise} \end{cases}$$

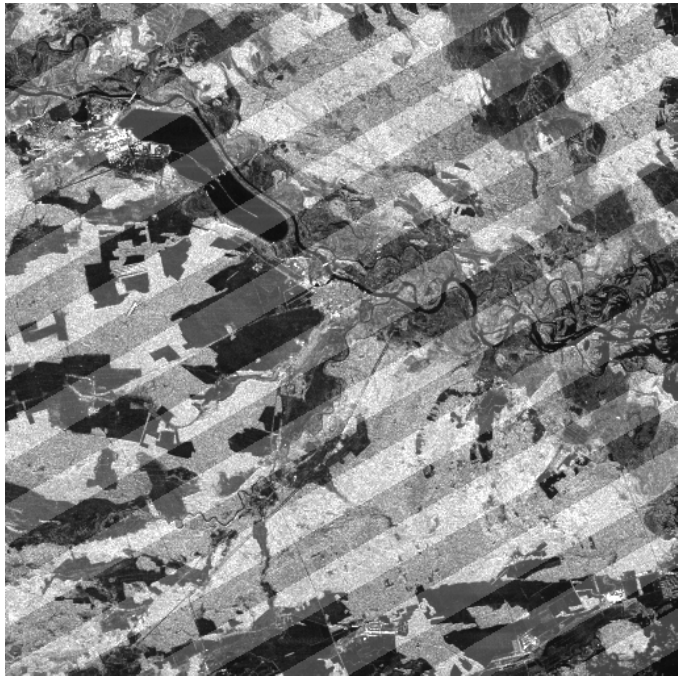
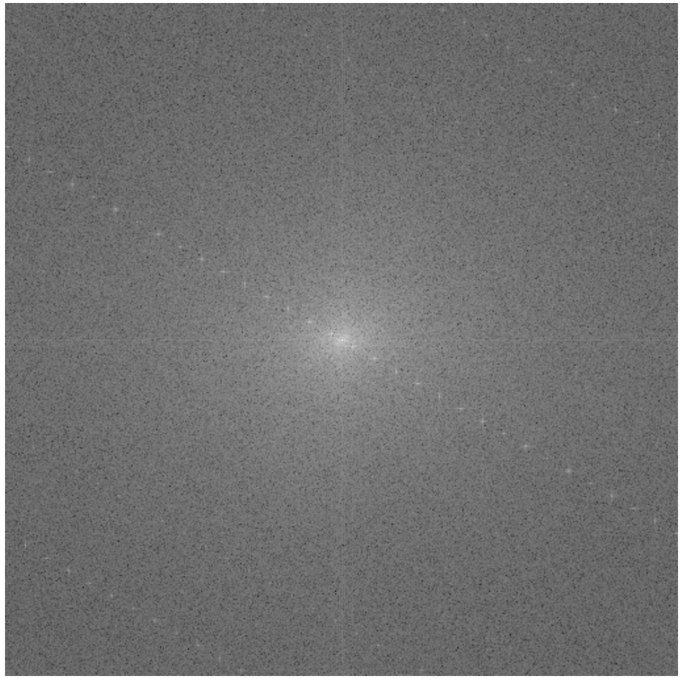
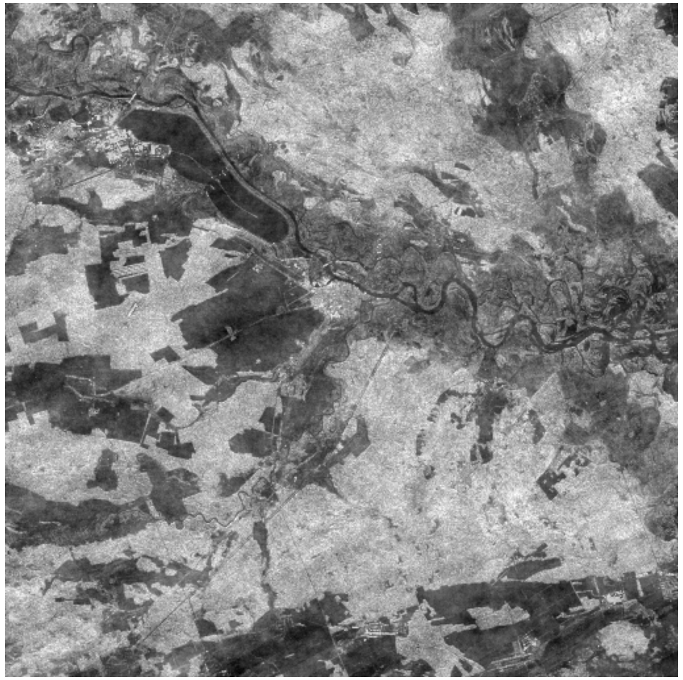


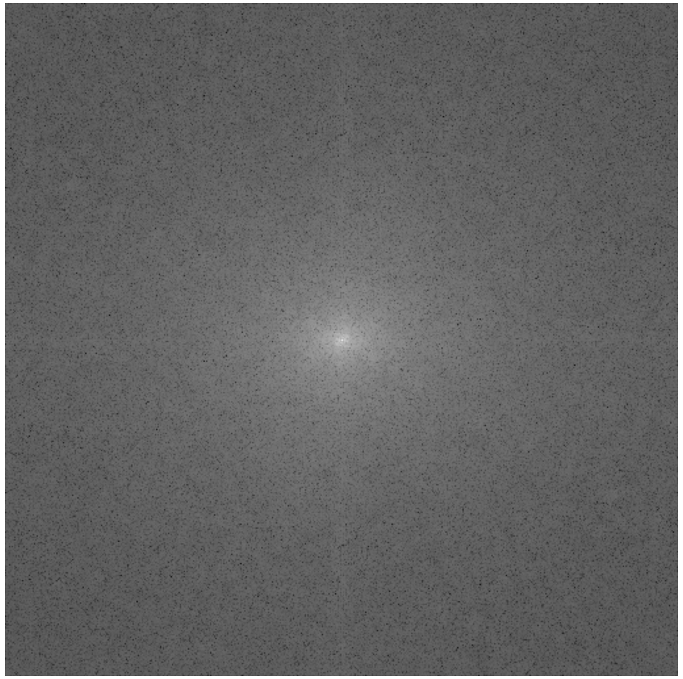
Image with periodic stripes



Its power spectrum



Filtered using median detector and median filter, T=3



Power spectrum of the filtered image

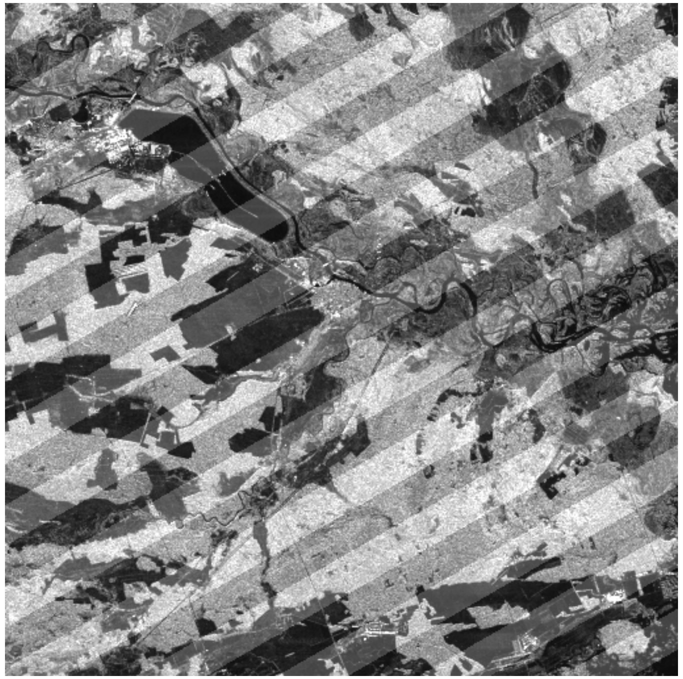
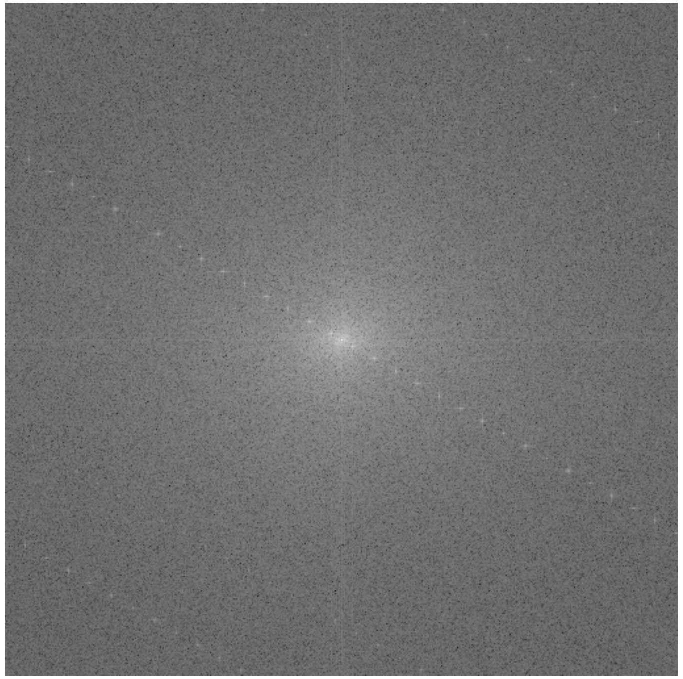
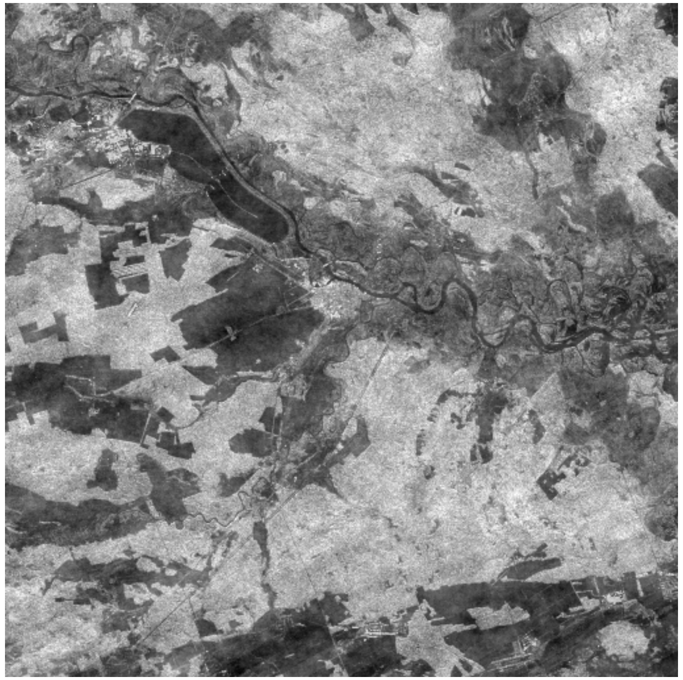


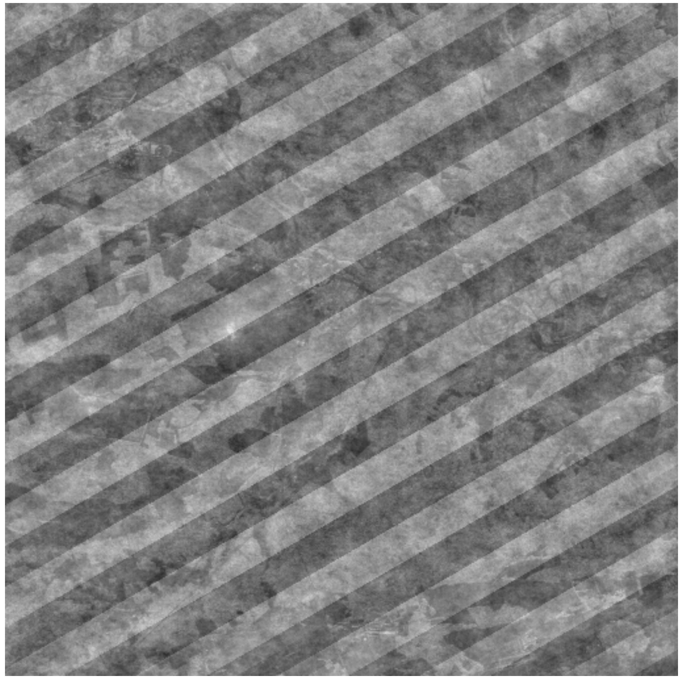
Image with periodic stripes



Its power spectrum



Filtered using median detector and median filter 11x11, T=3



Difference between noisy and filtered images

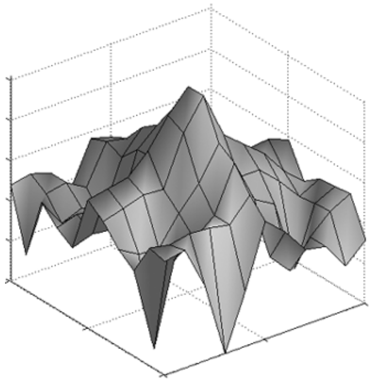
Gaussian Median Notch Filter

I. Aizenberg and C. Butakoff, "A Windowed Gaussian Notch Filter for Quasi-Periodic Noise Removal", Image and Vision Computing, vol. 26, No 10, October 2008, pp. 1347-1353

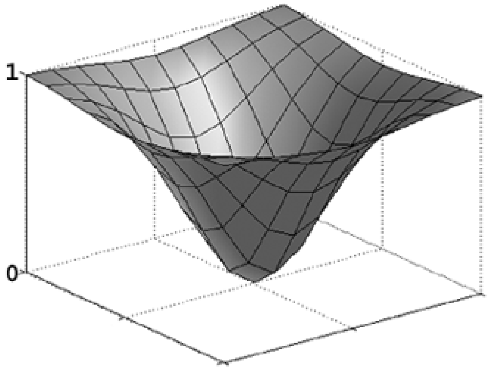
- This filter should be used in conjunction with median based peak detector (slide 33)
- The m x n vicinity of the peak is multiplied component-wise by the following surface:

$$G(u,v) = 1 - Ae^{-B\left[\left(u^2 + \left\lfloor \frac{n-1}{2} \right\rfloor^2\right) + \left(v^2 + \left\lfloor \frac{m-1}{2} \right\rfloor^2\right)\right]}$$

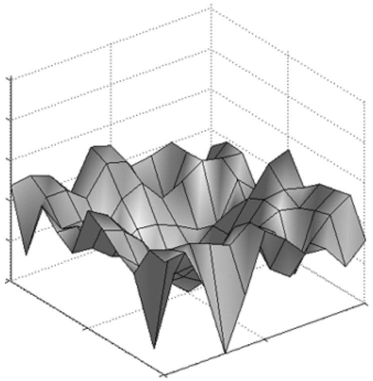
$$u = -\left\lfloor \frac{n}{2} \right\rfloor, \ldots, \left\lfloor \frac{n}{2} \right\rfloor; v = -\left\lfloor \frac{m}{2} \right\rfloor, \ldots, \left\lfloor \frac{m}{2} \right\rfloor; B < 1$$



$\otimes$



$=$



Peak

Gaussian-like Surface G

Filtered Peak

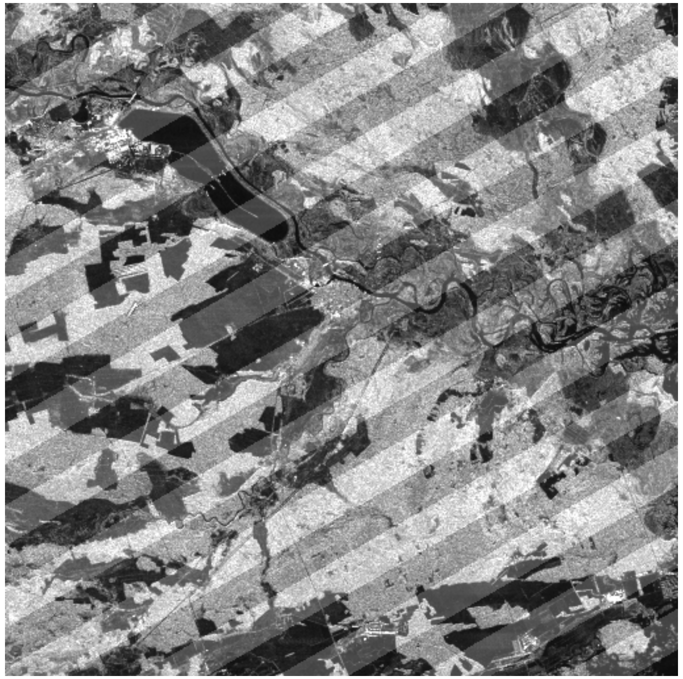
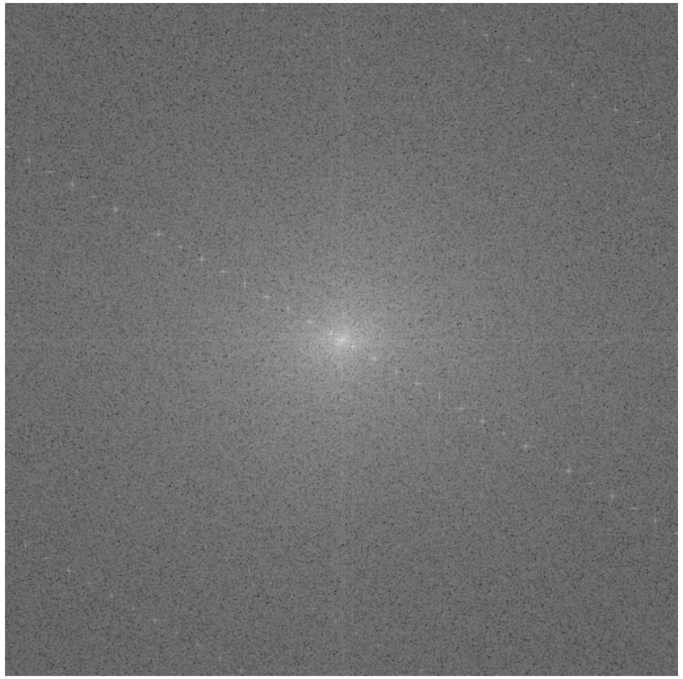
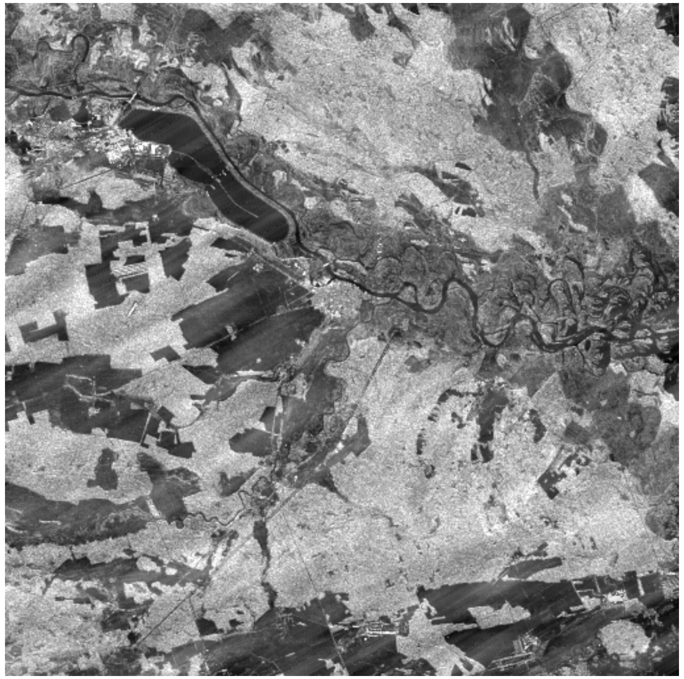


Image with periodic stripes

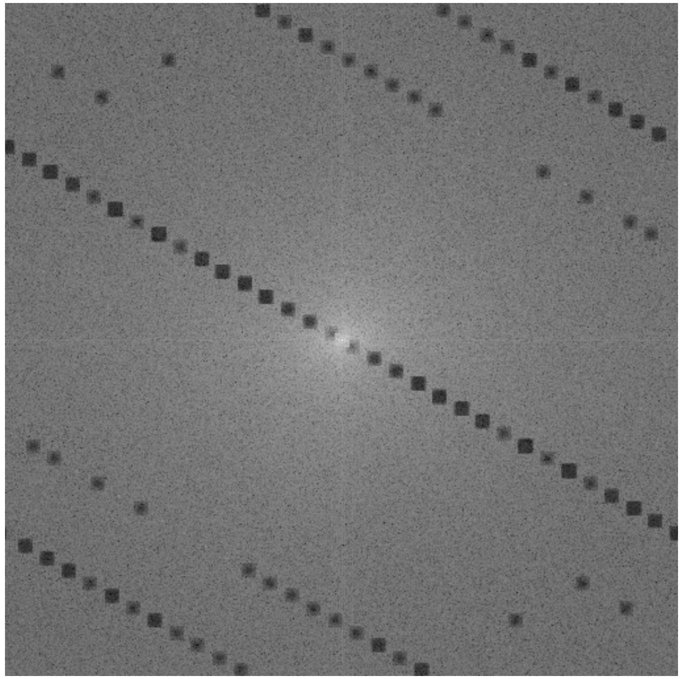
Its power spectrum





Filtered using Gaussian Median Notch filter 11x11, *T*=5

Power spectrum of the filtered image



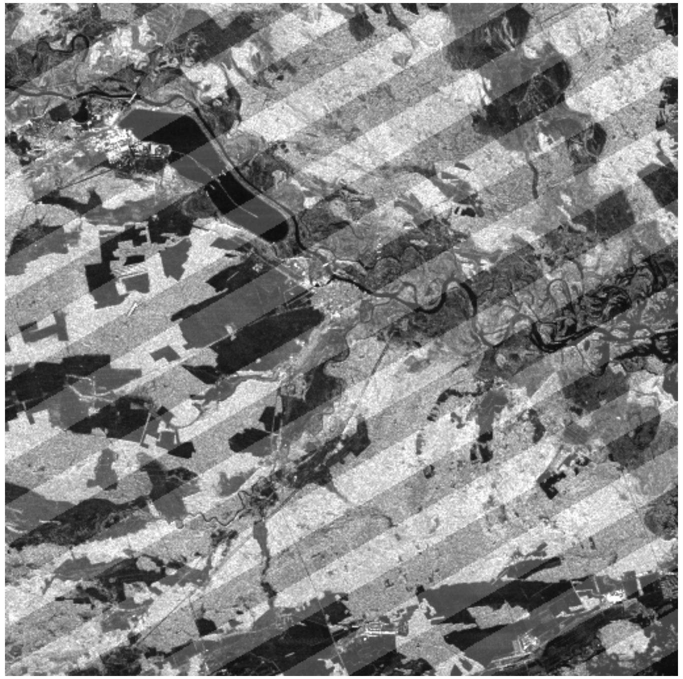
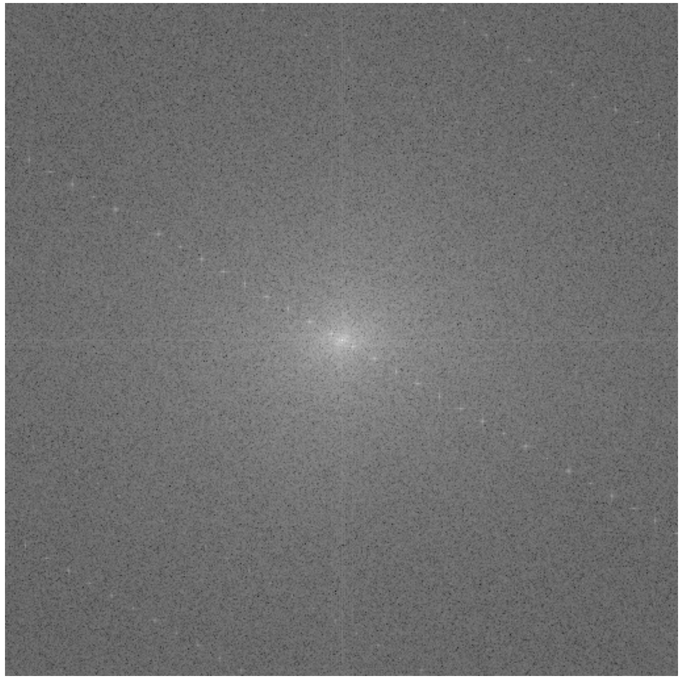
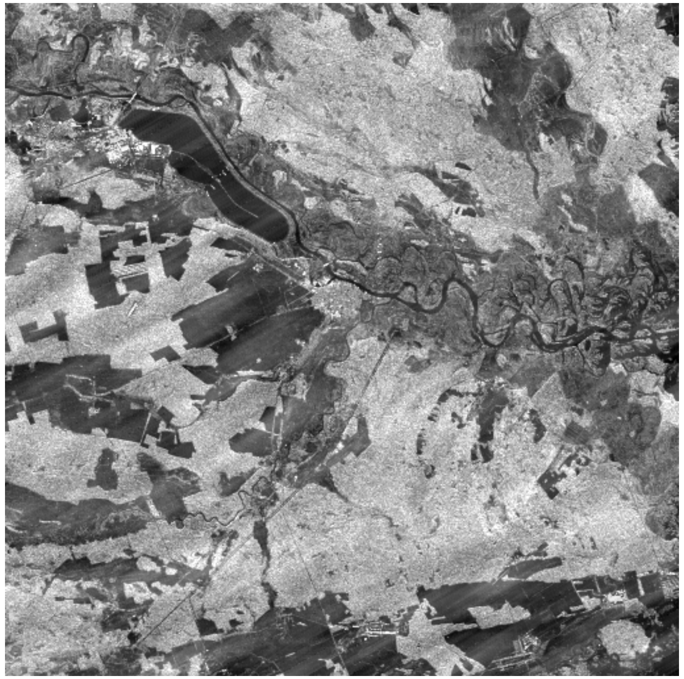


Image with periodic stripes

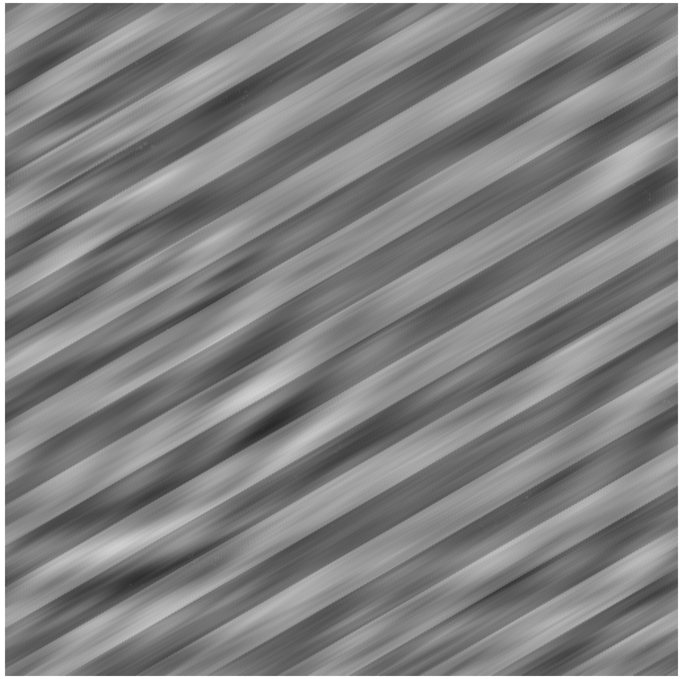
Its power spectrum





Filtered using Gaussian Median Notch filter 11x11, *T*=5

Difference between noisy and filtered images



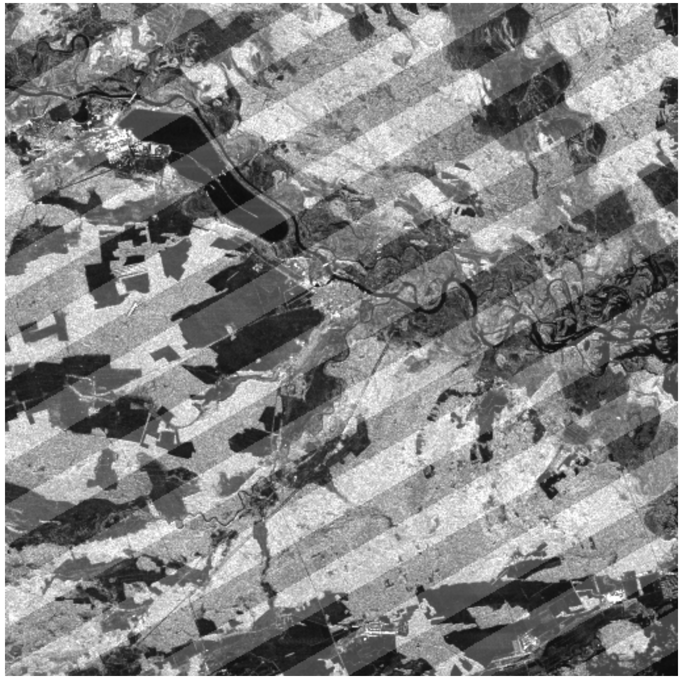
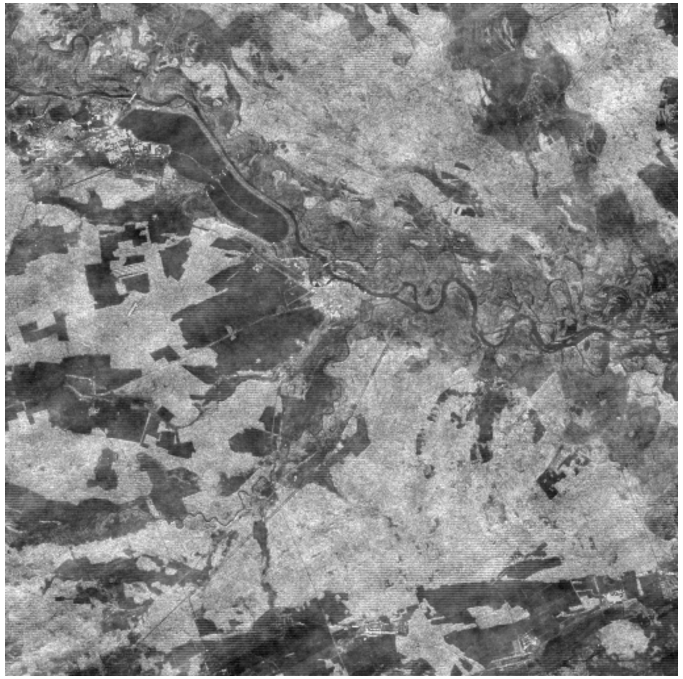
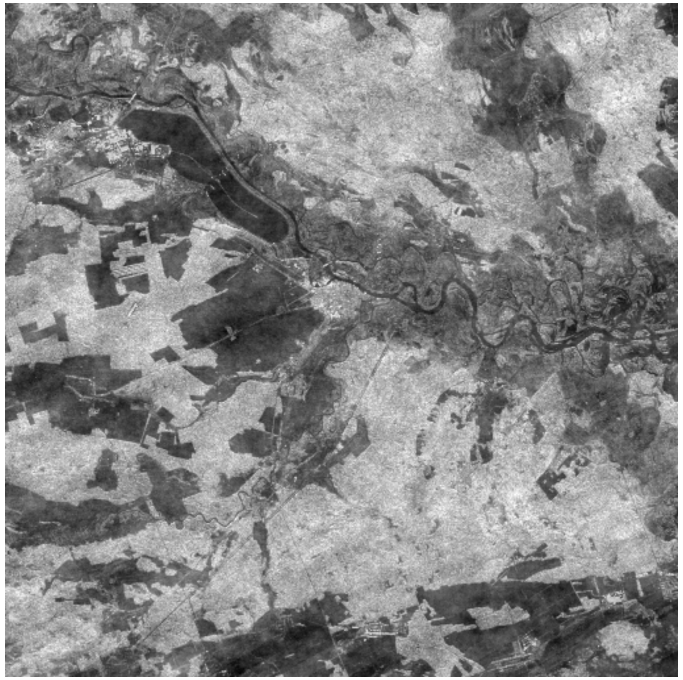


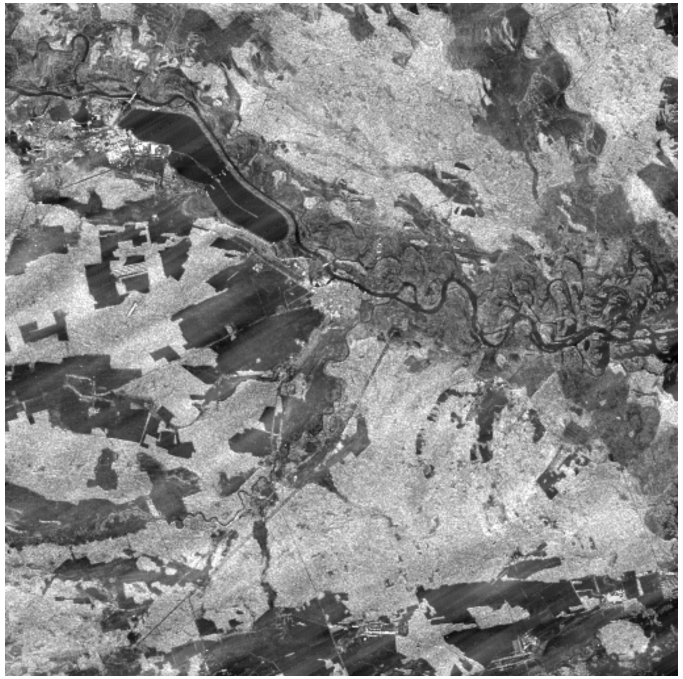
Image with periodic stripes



Thresholding, *T*=1,5



Median filter, 11x11 window, *T*=3



Gaussian Median Notch filter, 11x11 window, *T*=5

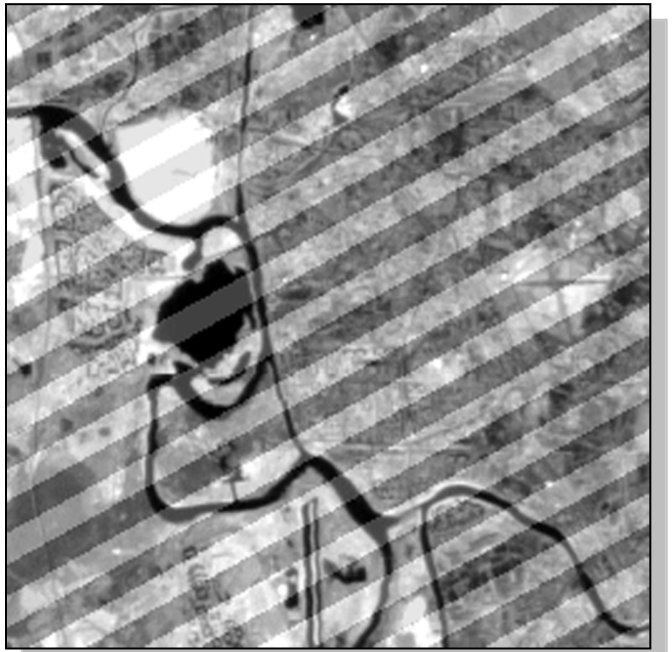


Image with periodic stripes

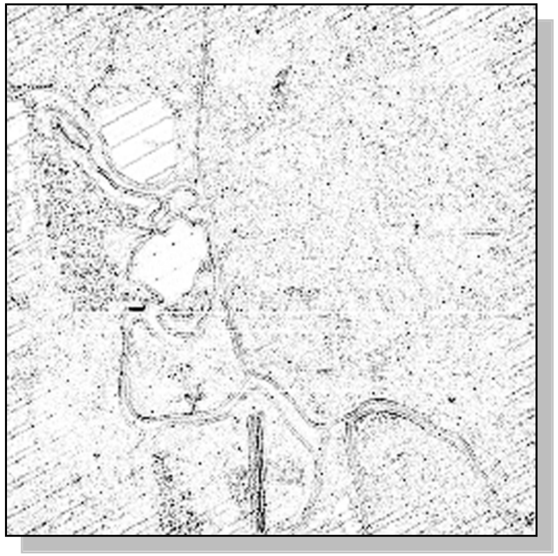
Thresholding, T=1.5

Median filter, 11x11 window, T=4

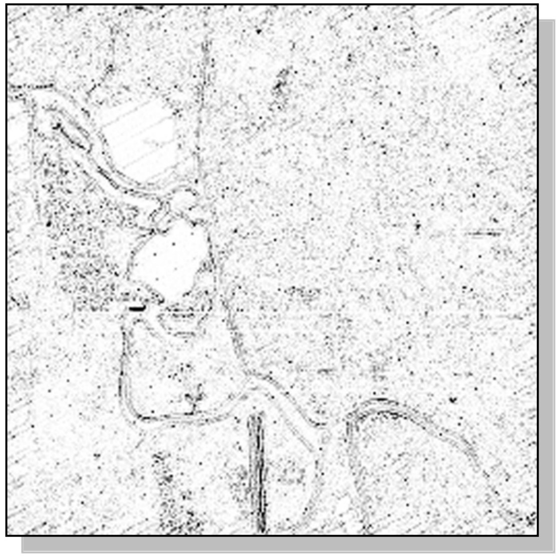
Gaussian Median Notch filter, 11x11 window, T=6

Laplacian Edge Detection

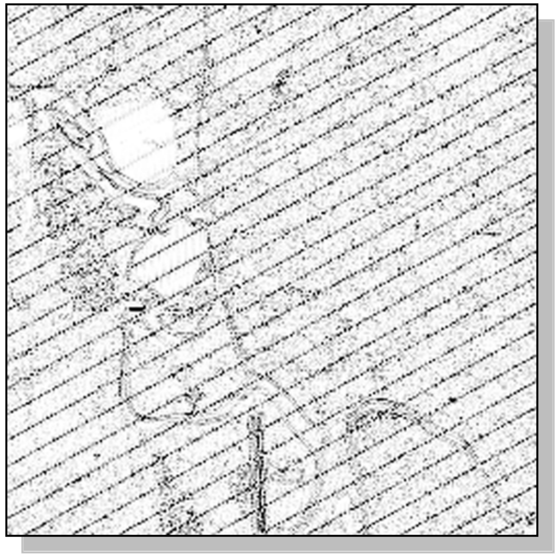
Median



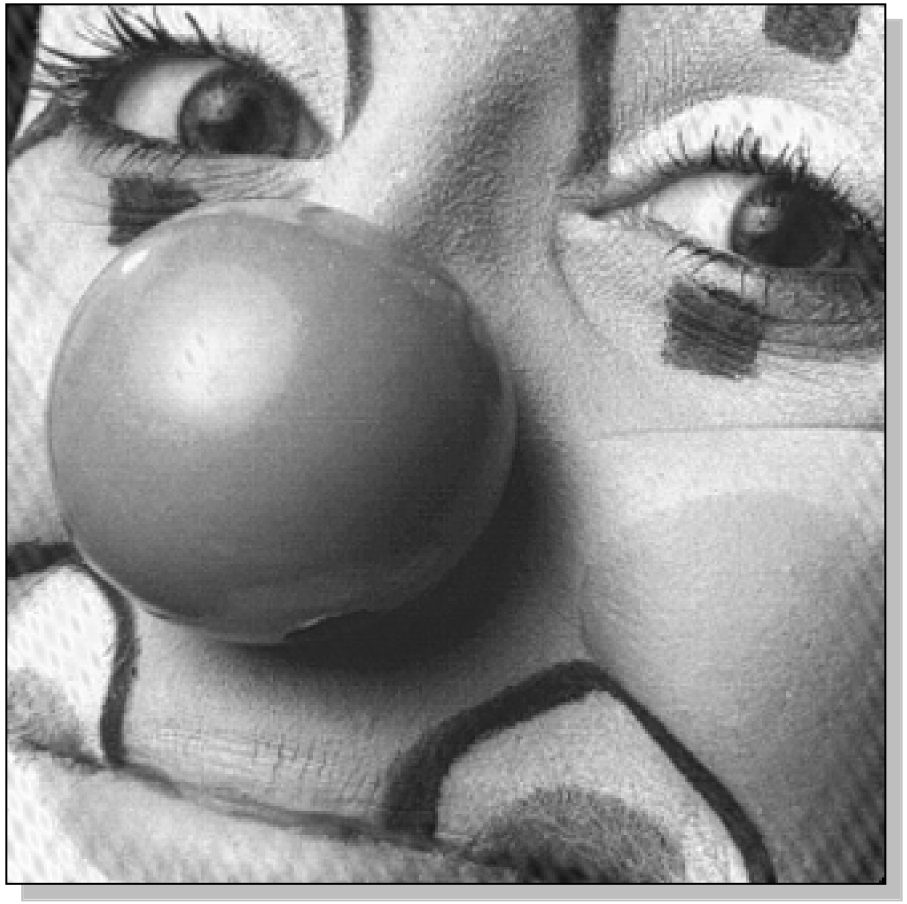
Gaussian Median Notch



Thresholding



Clown



Gaussian Median Notch Filter,
11x11 window, T=8

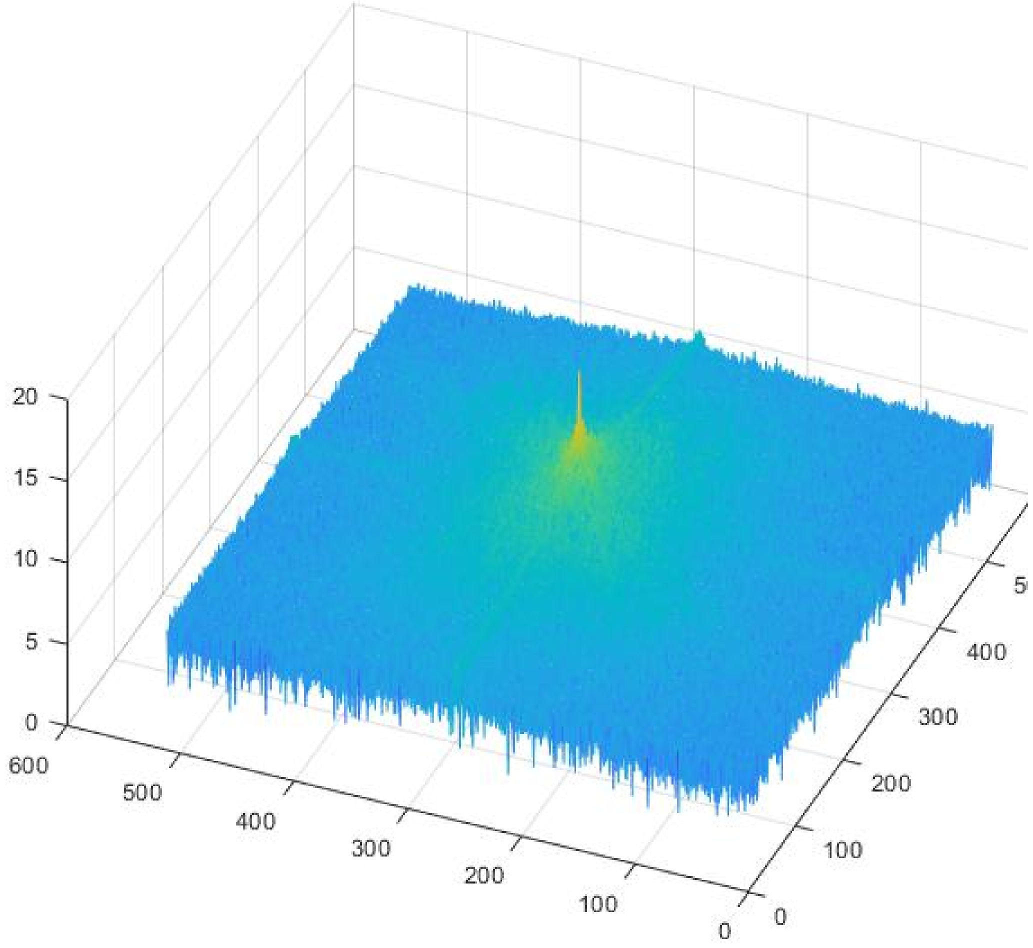
Gaussian Median Notch Filter, 11x11 window, *T*=11

Periodic Noise

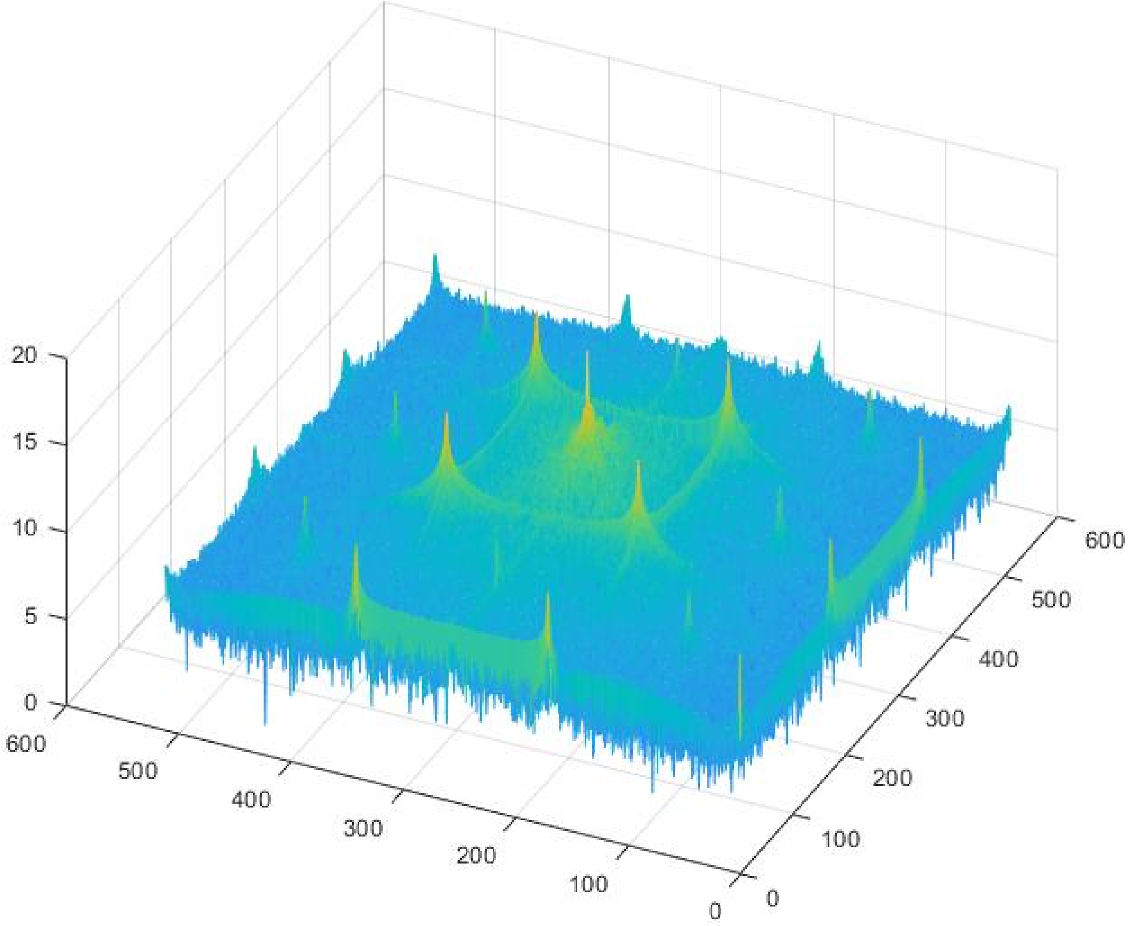
Original Image

Image corrupted by periodic noise (moire)

Periodic Noise

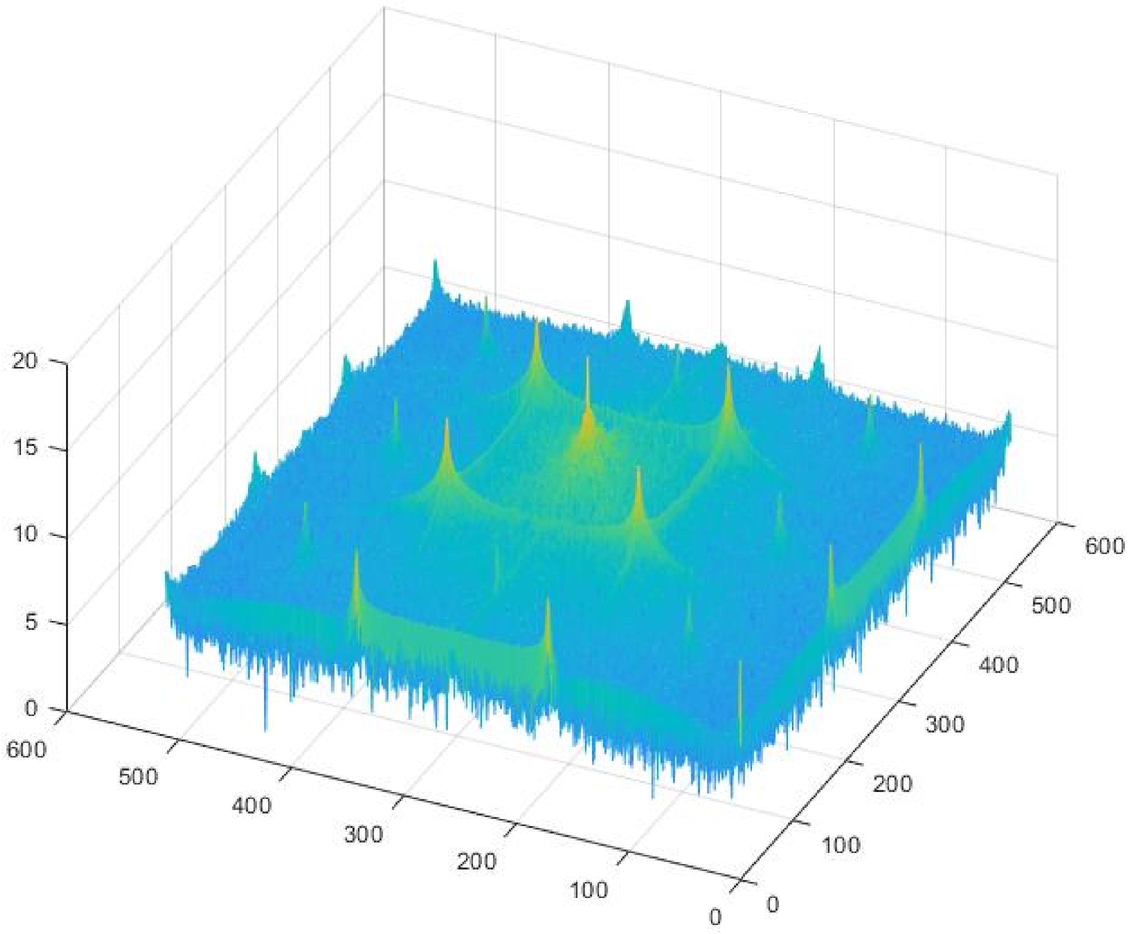


Fourier Power Spectrum of the Original Image

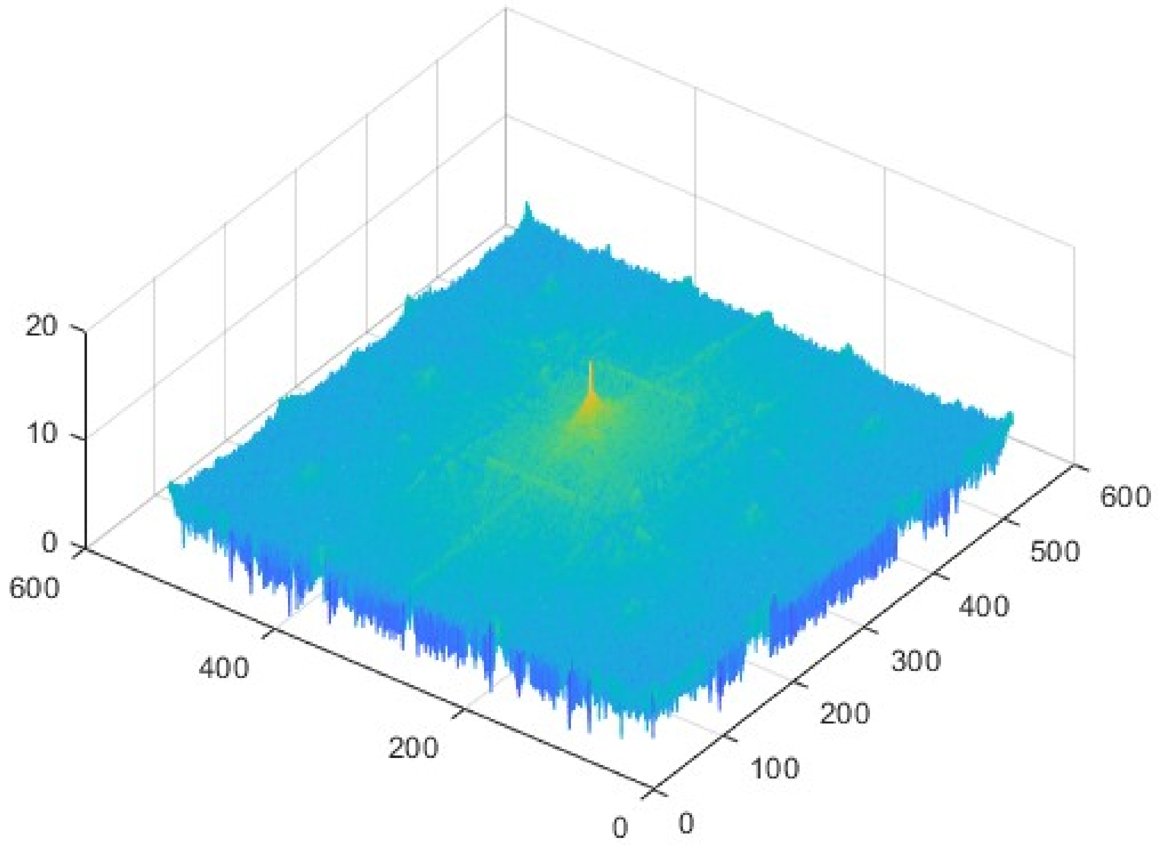


Fourier power spectrum of the Image corrupted by periodic noise (moire)

Periodic Noise: Gaussian Median Notch Filter



Fourier power spectrum of the Image corrupted by periodic noise (moire)



Fourier power spectrum of the image filtered using Gaussian Median Notch Filter, window 11x11, T=5, B=0.1

Periodic Noise: Gaussian Median Notch Filter

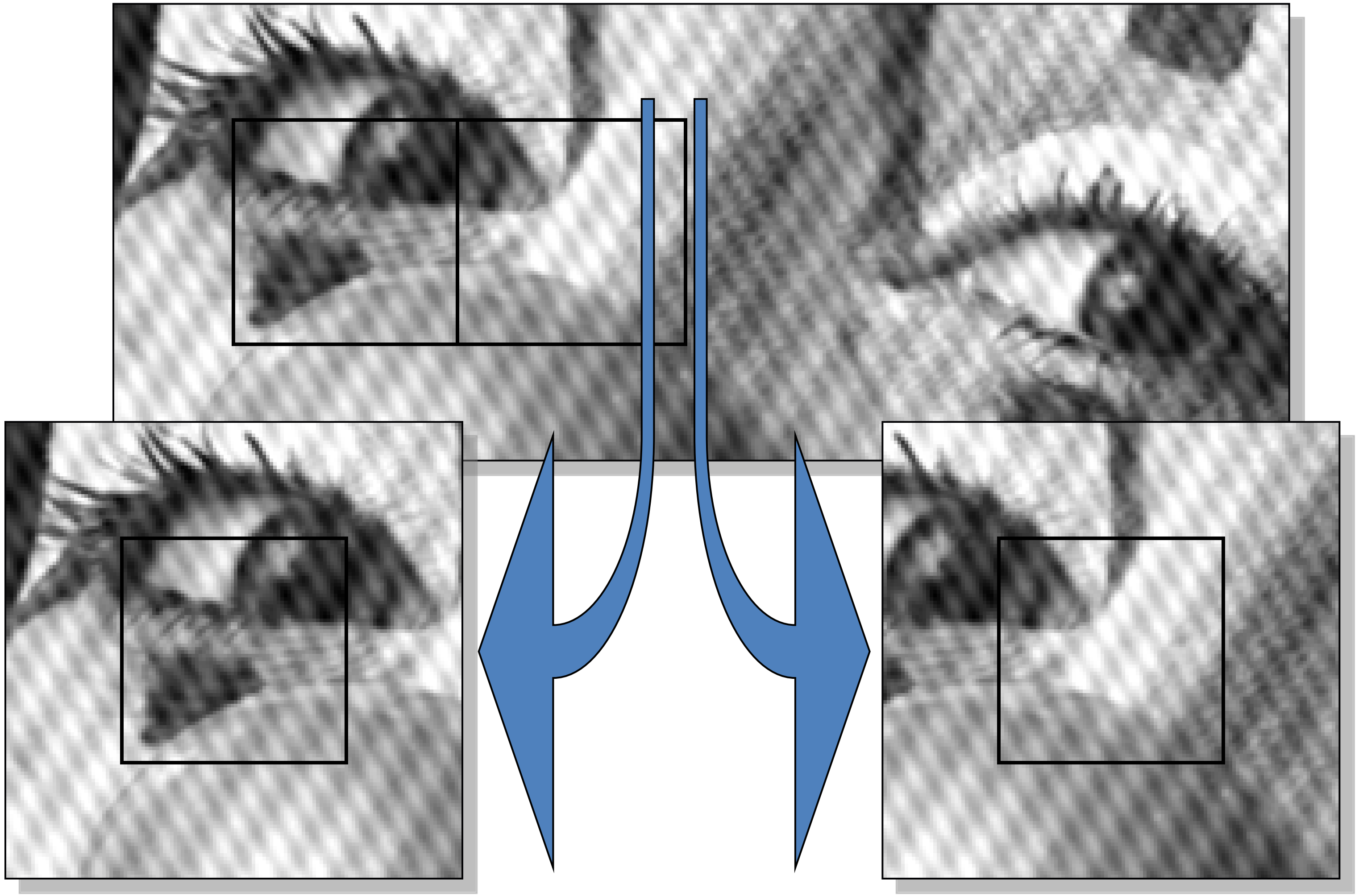
Image corrupted by periodic noise (moire)

Image filtered using Gaussian Median Notch Filter, window 11x11, T=5, B=0.1

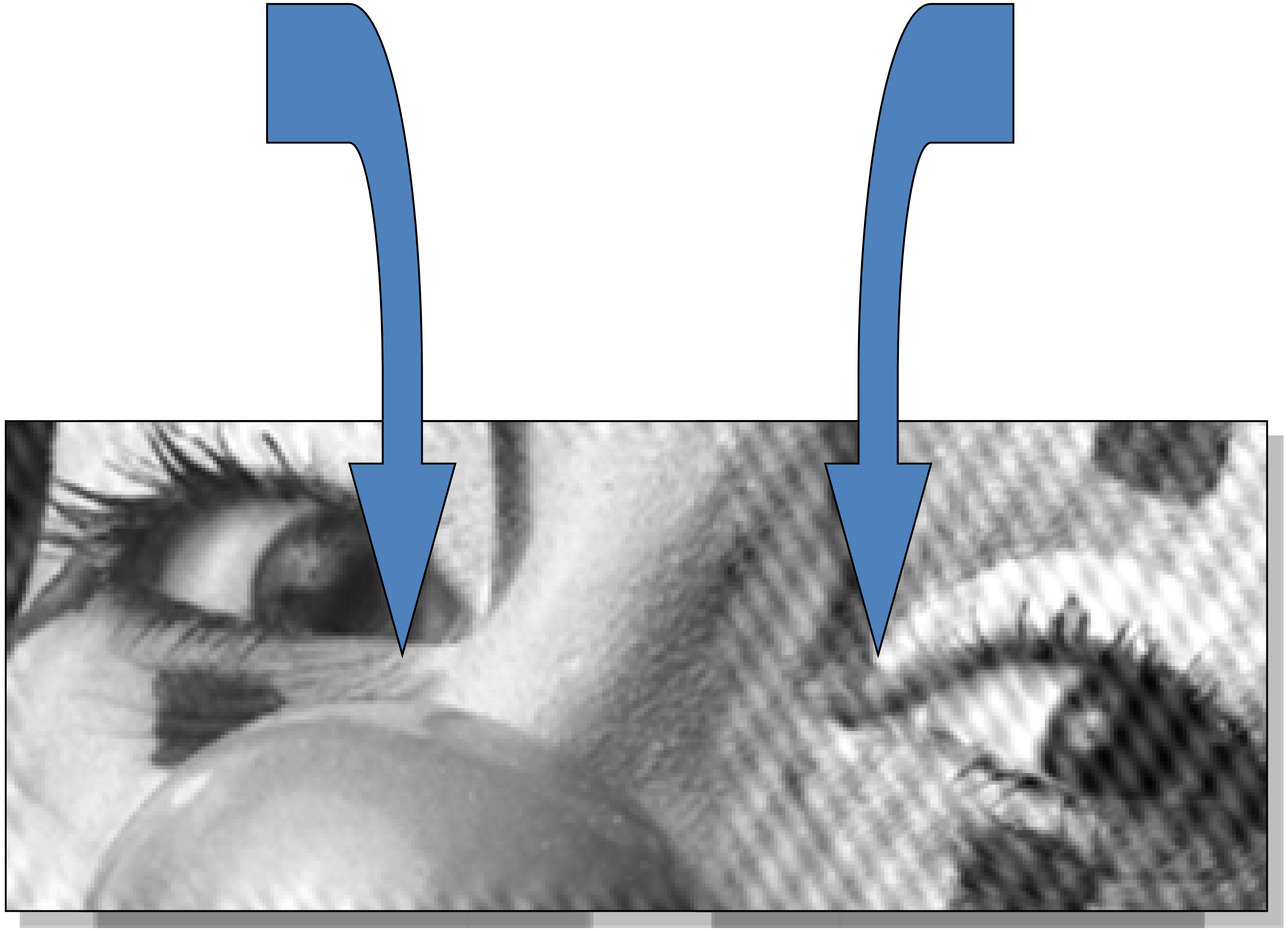
Filtering With Overlapping Windows

- If the image sizes are not the power of 2 (the limitation of FFT) it is better to filter it by blocks instead of extending to the closest power of 2.
- When the noise is non-uniform (quasi-periodic) there’s a chance that in smaller blocks it should be uniform.

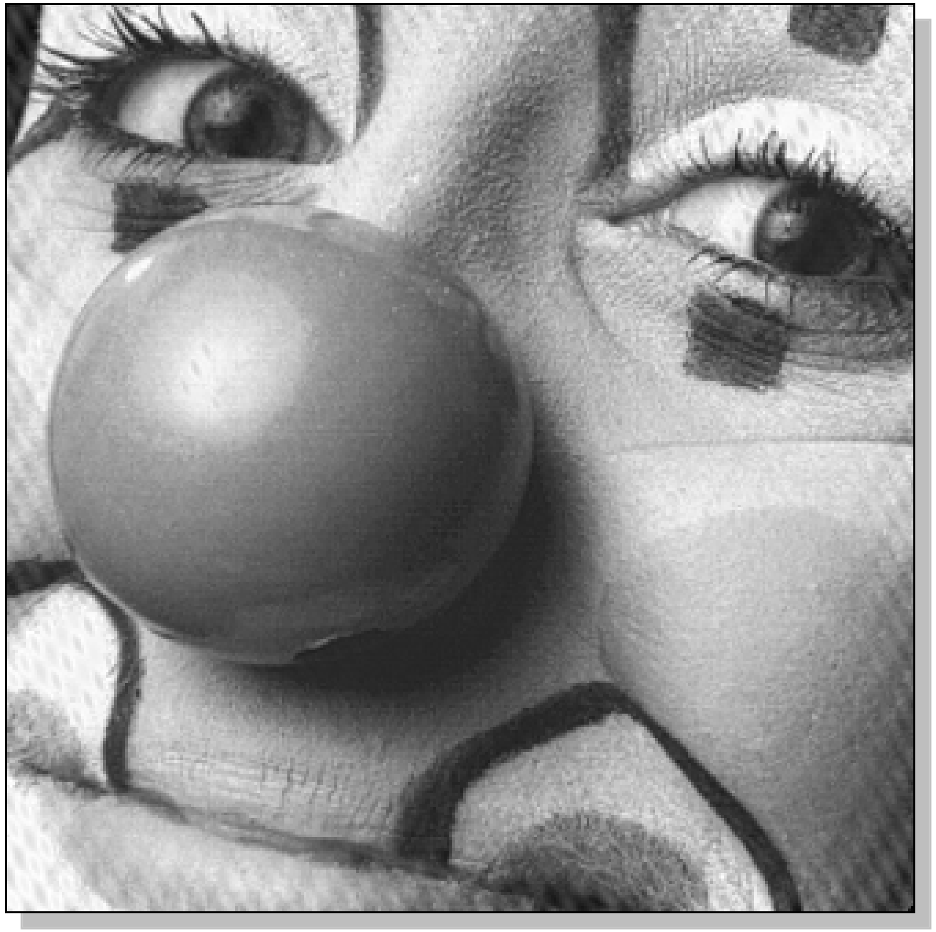
Block Extraction



Block Integration



Processed by blocks



Processed as a whole